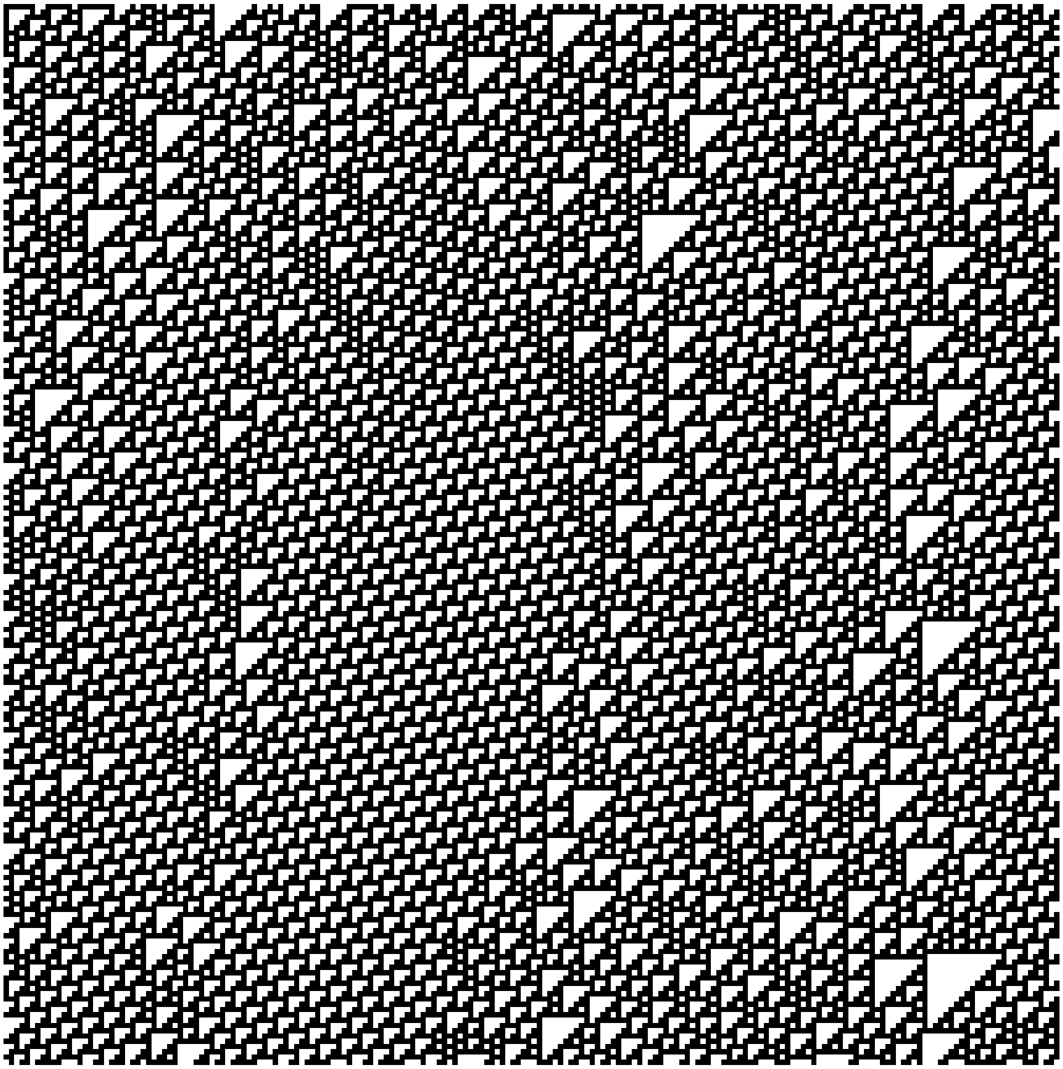
Cellular Automata

Jarkko Kari

Spring 2022



University of Turku

Contents

1 Preliminaries

1.1 Introduction

A cellular automaton is a discrete dynamical system that consists of a regular network of finite state automata (cells) that change their states depending on the states of their neighbors, according to a local update rule. All cells change their state simultaneously, using the same update rule. The process is repeated at discrete time steps. It turns out that amazingly simple update rules may produce extremely complex dynamics when applied in this fashion. A well known example is the *Game-of-life* by John Conway. Cellular automata are

- discrete in both space and time,
- homogeneous in space and time (same update rule at all cells at all times),
- local in their interactions.

Many processes in nature are governed by local and homogeneous underlying rules, which makes them amenable to modeling and simulation using cellular automata. For example, fluid dynamics can be modeled by moving point particles in a regular lattice, and the local update rule is designed to simulate particle collisions. Some of the most extensively investigated concepts in cellular automata theory such as reversibility and conservation laws are motivated by physics.

Cellular automata are also mathematical models for massively parallel computation. Simple update rules can make the cellular automaton computationally universal, that is, capable of performing arbitrary computation tasks. Above mentioned Game-of-life is a good example. This point of view raises interesting questions concerning the computational aspects of cellular automata.

A combination of the two viewpoints above (computational universality and modeling natural processes) have made cellular automata a useful theoretical tool in the study of computation in nature and the physical aspects and physical limits of computation.

These notes cover the basic theory of cellular automata. The most extensively used mathematical tool is topology. It namely turns out to be a very natural and fruitful approach to consider cellular automata as continuous functions on a compact metric space. This makes cellular automata theory part of the field of topological dynamics, or more specifically, symbolic dynamics. Since only elementary topology is needed, no prior mathematics courses in topology are required: the notes contain a review of all the topology and symbolic dynamics that is needed.

Another tool that we use is the theory of computation and computability. We are often interested in algorithmic questions related to cellular automata, and in many cases these questions turn out to be undecidable. A short review of computation theory, including universality and (un)decidability is included to help students who have no familiarity with this topic.

We start the notes with basic definitions and several examples of interesting cellular automata. We then continue with classical results related to injectivity and surjectivity. Chapters that follow (not necessarily in this order) discuss linear (additive) cellular automata, reversibility, limit sets, classifications of cellular automata, universality, conservation laws in cellular automata, topological dynamics of cellular automata, algorithmic questions, etc.

1.2 First example: Game-of-life

We start with a well-known example, *Game-of-life*, invented by John Conway in 1970. It is a cellular automaton that consists of an infinite grid of square cells — like an infinite graph paper — where each square is colored white or black. The color is called the state of the cell. We say that a black cell is alive while a white cell is not. A coloring of the entire grid is called a configuration of *Game-of-life*.

There is a simple local update rule according to which the cells change their states. The new state of a cell only depends on the current states of the cell itself and its eight nearest neighbors:

- A living cell stays alive if and only if there are exactly two or three living cells among the eight surrounding cells. Fewer than two living neighbors causes death by isolation, more than three living neighbors by overcrowding.
- A non-living cell becomes alive if it has precisely three living neighbors — each organism has three parents!

All cells use the same update rule, and all cells change their states simultaneously. This changes the coloring of the grid, i.e. the configuration changes into a new one. The process is then repeated over and over again, which creates a time evolution of the system. Figure 1 shows an example of five consecutive generations of cells.

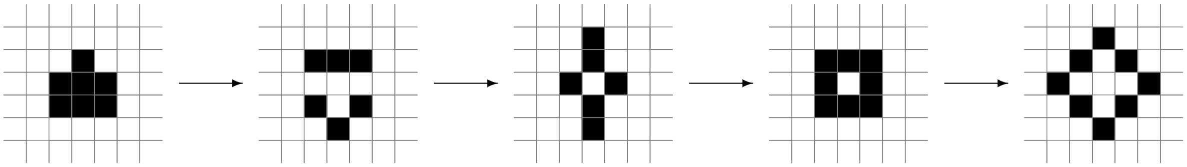


Figure 1: Five steps of a time evolution in Conway's *Game-of-life*.

Game-of-life is remarkable because the local update rule is extremely simple, but the long-time behavior of configurations is unpredictable. In the following the term "finite pattern" refers to a configuration in which the number of living cells is finite. Conway showed that it is undecidable if a given finite pattern eventually dies completely out. In other words, there is no (and never will be any as its existence is a logical contradiction) a computer program that takes as input a finite pattern and always correctly determines if the input pattern eventually dies out.

Over the years *Game-of-life* enthusiasts have compiled a vast library of patterns with various behaviors. The following terminology is used for various categories of objects:

- *still life*: a fixed point pattern. The update rule keeps each cell unchanged. The simplest non-empty still life is the *block*, a two-by-two block of living cells. Another still life is shown in Figure 2(a).

- *oscillator*: temporally periodic pattern. The update rule may change the pattern but after some number of steps the original pattern reappears in the same location and orientation. Still life is a special type of oscillator. The smallest oscillator is the *blinker* consisting of three living cells in a line. Another oscillator with period two is shown in Figure 2(b).

- *spaceship*: a pattern that after some number of steps reappears, possibly in a different location of the grid. A particular spaceship called *glider* is shown in Figure 2(c). An oscillator is a stationary spaceship that does not move.

- *gun*: a finite pattern that — like an oscillator — periodically returns back to the initial state, but in addition, emits spaceships. A *glider gun* emitting gliders is shown in Figure 2(d).

Objects from different categories emerge when *Game-of-life* is started in a random initial configuration. During the evolution the objects interact with each other through collisions with gliders and other moving structures. Collisions create new objects which in turn participate in interactions, leading to extraordinary complexity.

1.3 Basic Definitions

This chapter introduces the most basic definitions and notations. Throughout these notes, abbreviation CA refers to *cellular automata* (plural) or *cellular automaton* (singular).

Let d be a positive integer. A d-dimensional *cellular space* is $\mathbb{Z}^d$. Elements of $\mathbb{Z}^d$ are called *cells*. Let S be a finite *state set*. Elements of S are called *states*. A *configuration* of a d-dimensional CA with state set S is a function

$$c : \mathbb{Z}^d \longrightarrow S$$

that assigns a state to each cell. The state of cell $\vec{n} \in \mathbb{Z}^d$ is $c(\vec{n})$. A configuration should be understood as an instantaneous description, or a snapshot, of all the states in the system of cells at some moment of time. Most frequently we consider one- and two-dimensional spaces, in which cases the cells form a line indexed by $\mathbb{Z}$ or an infinite checker board indexed by $\mathbb{Z}^2$, respectively.

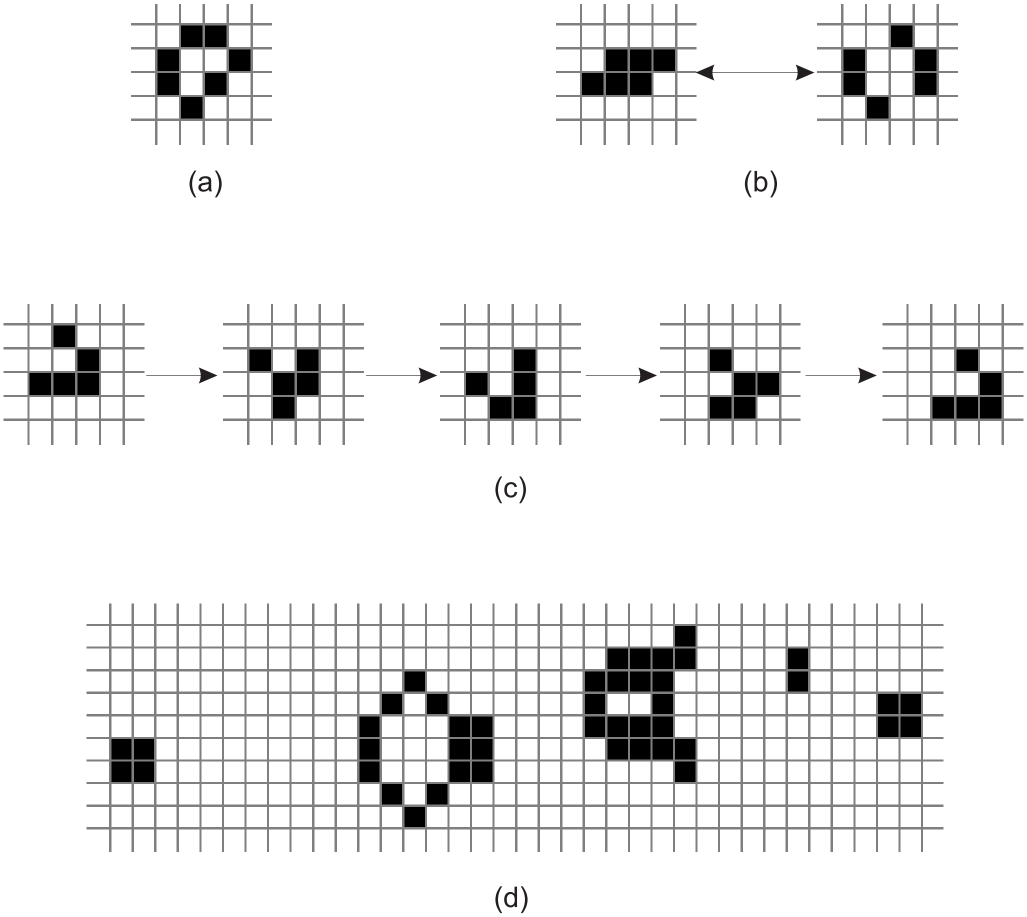


Figure 2: Sample Game of Life objects: (a) still life, (b) oscillator, (c) glider, (d) glider gun.

We adapt the common mathematical notation that the set of functions from set A into set B is denoted by B^A. So the set of all configurations is $S^{\mathbb{Z}^d}$. In the one-dimensional case $d = 1$ the set of configurations is $S^{\mathbb{Z}}$, the set of functions $\mathbb{Z} \longrightarrow S$.

A d-dimensional *neighborhood vector* (of size m) is a tuple

$$N = (\vec{n}_1, \vec{n}_2, \ldots, \vec{n}_m) \tag{1}$$

where each $\vec{n}_i \in \mathbb{Z}^d$ and $\vec{n}_i \neq \vec{n}_j$ for all $i \neq j$. The elements $\vec{n}_i$ specify the relative locations of the neighbors of each cell: Cell $\vec{n} \in \mathbb{Z}^d$ has m *neighbors* $\vec{n} + \vec{n}_i$ for $i = 1, 2, \ldots, m$.

The *local update rule* (or the *local rule*, the *update rule*, or simply the *rule*) of a CA with state set S and size m neighborhood is a function

$$f : S^m \longrightarrow S$$

that specifies the new state of each cell based on the old states of its neighbors. If the neighbors of a cell have states $s_1, s_2, \ldots, s_m$ then the new state of the cell is $f(s_1, s_2, \ldots, s_m)$.

In cellular automata all cells use the same rule, and the rule is applied at all cells simultaneously. This causes a global change in the configuration: Configuration c is changed into configuration c' where for all $\vec{n} \in \mathbb{Z}^d$

$$c'(\vec{n}) = f[c(\vec{n}+\vec{n}_1), c(\vec{n}+\vec{n}_2), \dots, c(\vec{n}+\vec{n}_m)]. \tag{2}$$

The transformation $c \mapsto c'$ is the global *transition function* of the CA. It is a function

$$G : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}.$$

Function G is our main object of study. Typically, function G is iterated, i.e. applied repeatedly, which produces a time evolution

$$c \mapsto G(c) \mapsto G^2(c) \mapsto G^3(c) \mapsto \dots$$

of the system. Here c is the initial configuration of the evolution, and the sequence

$$orb(c) = c, G(c), G^2(c), G^3(c), \dots$$

is the *orbit* of c. Time refers to the number of applications of G performed: Each application of G takes one time step, so $G^t(c)$ is the configuration at time t, for all $t = 0, 1, 2, \dots$.

Sometimes we also consider *two-way infinite orbits*, i.e. sequences

$$\dots, c_{-2}, c_{-1}, c_0, c_1, c_2, \dots$$

of configurations where $G(c_i) = c_{i+1}$ for all $i \in \mathbb{Z}$. Here time t flows through all integers and there is no initial configuration.

In summary: To specify a CA one needs to specify the following items (some of which may be clear from the context):

- the dimension $d \in \mathbb{Z}_+$,
- the finite state set S,
- the neighborhood vector $N = (\vec{n}_1, \vec{n}_2, \dots, \vec{n}_m)$, and
- the local update rule $f : S^m \longrightarrow S$.

We therefore formally define the corresponding CA to be the 4-tuple $A = (d, S, N, f)$. The global transition function determined by these items according to (2) will be denoted by $G[A]$, or simply by G when the CA A is clear from the context. Any function G that is the transition function of some CA is called a *CA function*.

We usually identify a CA function G with the CA that determines it in the sense that we talk about cellular automaton G. Strictly speaking, however, the same function G is determined by different cellular automata (4-tuples). We say that two CA A and B are *equivalent* if $G[A] = G[B]$. Clearly equivalent CA have the same dimension d and state set S but they may differ in their neighborhood vectors. However, we see in the following section that there is a unique equivalent CA whose neighborhood vector is minimal in the sense that it is included in the neighborhoods of all equivalent CA. Other equivalent CA can only have additional "dummy" neighbors that have no influence on the next state.

Example 1. (*xor*) Let $d = 1$, $S = \{0, 1\}$, $N = (0, 1)$ and $f : \{0, 1\}^2 \longrightarrow \{0, 1\}$ be

$$f(a, b) = a + b \pmod 2.$$

The cells form a line, indexed by $\mathbb{Z}$. Each cell changes its state by adding the state of its right neighbor to its own old state modulo 2. This is known as the "exclusive or" (xor) logic operation.

Consider, for example, the initial configuration c_0 where $c_0(0) = 1$ and $c_0(i) = 0$ for all $i \neq 0$, i.e. a single cell is in state 1. Then $c_1 = G(c_0)$ has $c_1(0) = c_1(-1) = 1$ and $c_2(i) = 0$ for all $i \neq -1, 0$. Continuing likewise, we get the time evolution $c_2 = G(c_1)$, $c_3 = G(c_2)$ etc. Figure 3 shows a diagram where we have drawn configurations as horizontal rows of states and depicted values 0 and 1 by white and black squares as we'll typically do in our examples. The topmost row shows the initial configuration c_0, and the following rows represent consecutive elements of the orbit $orb(c_0)$. Time increases downwards.

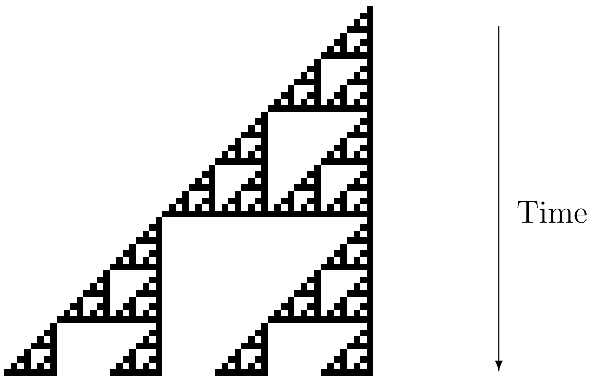


Figure 3: The space-time diagram of the *xor* CA of Example 1 starting from an initial configuration with a single cell in state 1.

□

A *space-time diagram* is a pictorial representation of an orbit, similar to the one shown in Example 1 above. In the case of one-dimensional CA configurations are drawn as horizontal lines of colors, each state represented by its own color. Configuration $G(c)$ is drawn under c, so time flows downwards. The topmost row represents the initial configuration. The space-time diagram of the orbit of c hence fills the lower half plane. In contrast, the space-time diagrams associated with two-way infinite orbits fill the whole plane since there is no initial time.

More generally, a space-time diagram of a d-dimensional CA is a $(d+1)$-dimensional "drawing" where d dimensions represent space and the additional dimension is used for time. Time gets values in $\mathbb{N}$ or $\mathbb{Z}$ depending on whether the diagram is for an orbit with an initial configuration or for a two-way infinite orbit. In the first case, the diagram is an element of $S^{\mathbb{Z}^d \times \mathbb{N}}$, and in the second case it belongs to $S^{\mathbb{Z}^d \times \mathbb{Z}}$.

Following terminology is used: A configuration c is

- a *fixed point* of G if $G(c) = c$.
- *(temporally) periodic* if $G^t(c) = c$ for some $t \in \mathbb{Z}_+$. Any t satisfying $G^t(c) = c$ is called a *period* of c and the smallest such t is the *least period* of c.
- *eventually fixed* if there is $n \in \mathbb{N}$ such that $G^{n+1}(c) = G^n(c)$, that is, $G^n(c)$ is a fixed point for some n.
- *eventually (temporally) periodic* if there is $n \in \mathbb{N}$ and $t \in \mathbb{Z}_+$ such that $G^{n+t}(c) = G^n(c)$, that is, $G^n(c)$ is periodic for some n.

Analogous terminology is used for orbits. A one- or two-way infinite orbit is a fixed point orbit if all configurations it contains are fixed points (i.e. the orbit consists of copies of the same fixed point configuration). It is periodic if it only contains temporally periodic configurations, it is eventually fixed if it contains some fixed point configuration and it is eventually periodic if it contains a temporally periodic configuration. See Figure 4 for illustrations of these concepts on two-way infinite orbits. The figure shows parts of phase spaces of some CA. A *phase space* is the infinite directed graph whose vertices are the configurations and from each configuration c there is exactly one outgoing edge leading to $G(c)$. Note that the phase space has uncountably many vertices, so we always show just a small portion of it, e.g. to plot some orbits as in Figure 4.

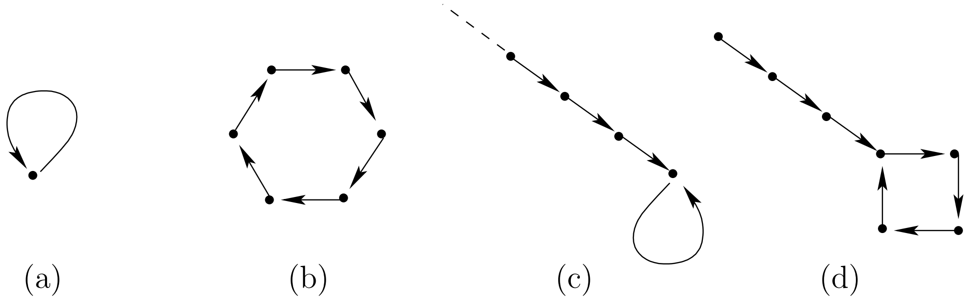


Figure 4: (a) a fixed point, (b) a periodic orbit, (c) an eventually fixed orbit and (d) an eventually periodic orbit.

As a final observation of this section we state the following simple fact:

Proposition 1 *If G and H are CA functions, so is their composition $G \circ H$.* □

1.4 Neighborhoods

Let $N = (\vec{n}_1, \vec{n}_2, \ldots, \vec{n}_m)$ be a d-dimensional neighborhood vector. For any $\vec{n} \in \mathbb{Z}^d$ we denote

$$N(\vec{n}) = (\vec{n} + \vec{n}_1, \vec{n} + \vec{n}_2, \ldots, \vec{n} + \vec{n}_m),$$

and for any $K \subseteq \mathbb{Z}^d$ we denote

$$N(K) = \{\vec{n} + \vec{n}_i \mid \vec{n} \in K \text{ and } i = 1, 2, \ldots, m\ \}.$$

In other words, $N(\vec{n})$ is the ordered sequence of the neighbors of cell $\vec{n}$, while $N(K)$ is the unordered set of neighbors of cells in K. In particular, $N(\{\vec{n}\})$ is the unordered set of neighbors of cell $\vec{n}$. Clearly $N = N(\vec{0})$, and $N(\{\vec{0}\})$ is the unordered set that contains the elements of the neighborhood vector N. The order of the elements in N is essentially irrelevant: it only matters as the order in which the m input values are given in the local rule $f : S^m \longrightarrow S$. So when specifying a CA it is enough to give the unordered version $N(\{\vec{0}\})$ of N, as long as we make the role of different neighbors clear in the description of the local rule.

In two-dimensional spaces the *von Neumann-* and the *Moore-* neighborhoods shown in Figure 5 are often used. *Game-of-life* has the Moore-neighborhood. We generalize the Moore-neighborhood and call the d-dimensional neighborhood M_r^d consisting of all

$$(k_1, k_2, \ldots, k_d) \in \mathbb{Z}^d \text{ where } |k_i| \leq r \text{ for all } i = 1, 2, \ldots, d$$

the *radius-r* neighborhood. It contains $(2r + 1)^d$ elements. We also generalize the von Neumann -neighborhood and call the d-dimensional neighborhood V_r^d consisting of

$$(k_1, k_2, \ldots, k_d) \in \mathbb{Z}^d \text{ where } \sum_{i=1}^{d} |k_i| \leq r$$

the radius-r von Neumann -neighborhood. The classical von Neumann and Moore neighborhoods of Figure 5 are then V_1^2 and M_1^2. Note that in the one-dimensional case $V_r^1 = M_r^1$.

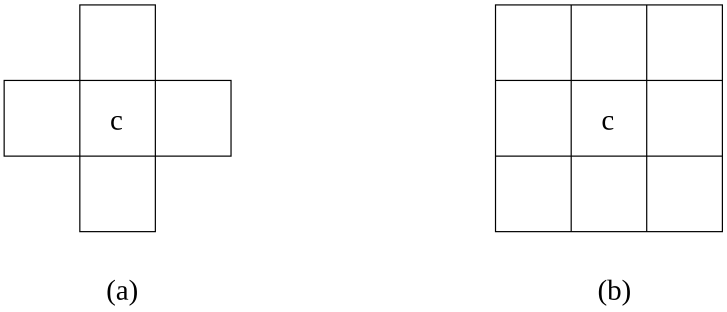


Figure 5: The (a) von Neumann and (b) Moore neighbors of cell c.

The following, small neighborhoods will be sometimes used: The *radius-*$\frac{1}{2}$ neighborhood consists of all $(k_1, k_2, \ldots, k_d) \in \mathbb{Z}^d$ where each $k_i \in \{0, 1\}$, and the radius-$\frac{1}{2}$ von Neumann -neighborhood consists of all $(k_1, k_2, \ldots, k_d) \in \mathbb{Z}^d$ where at most one k_i is 1 and all others are 0. In the one-dimensional case these both consist of the cell and its immediate right neighbor. The *xor* CA of Example 1 has the radius-$\frac{1}{2}$ neighborhood.

Consider a CA with neighborhood vector $N = (\vec{n}_1, \vec{n}_2, \ldots, \vec{n}_m)$ and local rule $f : S^m \longrightarrow S$. We call $\vec{n}_j$ a *dummy* neighbor if $f(s_1, \ldots, s_m) = f(t_1, \ldots, t_m)$ whenever $s_i = t_i$ for all $i \neq j$. This means that the the j'th neighbor of a cell has no effect on the next state of that cell, and hence $\vec{n}_j$ can be removed from the neighborhood vector. We obtain an equivalent CA with $m - 1$ neighbors. Let us say a CA has *minimal neighborhood* if it has no dummy neighbors. By removing all dummy neighbors from any CA we obtain an equivalent CA that has minimal neighborhood. This minimal neighborhood CA is unique:

Proposition 2 *If A and B are equivalent CA and have minimal neighborhoods then $A = B$ (up to reordering the neighbors in the neighborhood vector).*

Proof. It is enough to show that the neighborhood vectors of A and B contain the same elements. Let $\vec{n}$ be an arbitrary element of the neighborhood vector of A, that is, cell $\vec{n}$ is a neighbor of cell $\vec{0}$ in A. Because A has no dummy neighbors there exist two configurations c and e such that $c(\vec{n}) \neq e(\vec{n})$, $c(\vec{k}) = e(\vec{k})$ for all $\vec{k} \neq \vec{n}$ and $c'(\vec{0}) \neq e'(\vec{0})$ where we have denoted $c' = G(c)$ and $e' = G(e)$ and G is the global transition function of A. Since A and B are equivalent, G is also the transition function of B. This means that $\vec{n}$ has to be a neighbor of $\vec{0}$ also in B, as otherwise we would have $c'(\vec{0}) = e'(\vec{0})$.

In the same way, every neighbor in B is also a neighbor in A. □

1.5 Elementary CA

Elementary CA are one-dimensional cellular automata with two states and radius-1 neighborhood: $d = 1$, $S = \{0, 1\}$, $N = (-1, 0, 1)$ and $f : S^3 \longrightarrow S$. They differ from each other only in the choice of the local rule f. There are 256 elementary CA because the number of different local rules $S^3 \longrightarrow S$ is $2^8 = 256$. Note, however, that some of the 256 elementary rules are identical up to renaming the states or reversing right and left, so the number of essentially different elementary rules is smaller, only 88.

Elementary rules were extensively studied and empirically classified by S.Wolfram in the 1980's. He introduced a naming scheme that has since become standard: Each elementary rule is specified by an eight bit sequence

$$f(111)\ f(110)\ f(101)\ f(100)\ f(011)\ f(010)\ f(001)\ f(000)$$

where f is the local update rule of the CA. The bit sequence is the binary expansion of an integer in the interval $0 \ldots 255$, called the *Wolfram number* of the CA.

Example 2. The 8 bit binary expansion of the decimal number 102 is 01100110 so the elementary CA with Wolfram number 102 has the local update rule

$$\begin{array}{llll} f(111)=0, & f(110)=1, & f(101)=1, & f(100)=0, \\ f(011)=0, & f(010)=1, & f(001)=1, & f(000)=0, \end{array}$$

This CA is equivalent to the *xor* CA of Example 1. □

Example 3.(*rule 110*) The 8 bit binary expansion of the decimal number 110 is 01101110 so the elementary CA with Wolfram number 110 has the local update rule

$$\begin{array}{llll} f(111)=0, & f(110)=1, & f(101)=1, & f(100)=0, \\ f(011)=1, & f(010)=1, & f(001)=1, & f(000)=0, \end{array}$$

This CA has become known since it was recently proved to be computationally universal. The cover page of these notes contains a snapshot of the space-time diagram of rule 110 started from a random initial configuration. □

Wolfram's numbering scheme is easily generalized to larger neighborhoods and state sets. One-dimensional, radius-r CA with k states is identified by a number that contains k^{2r+1} base-k digits.

S.Wolfram experimented in the 80's with elementary CA, and based on empirical observations of their behavior on random initial configurations he classified them into four classes. These are known as Wolfram classes of CA. The definitions are not mathematically rigorous, and more precise classifications (which we'll discuss later) have since been proposed. Wolfram defined the classes as follows:

(W1) Almost all initial configurations lead to the same uniform fixed point configuration,

(W2) Almost all initial configurations lead to a periodically repeating configuration,

(W3) Almost all initial configurations lead to essentially random looking behavior,

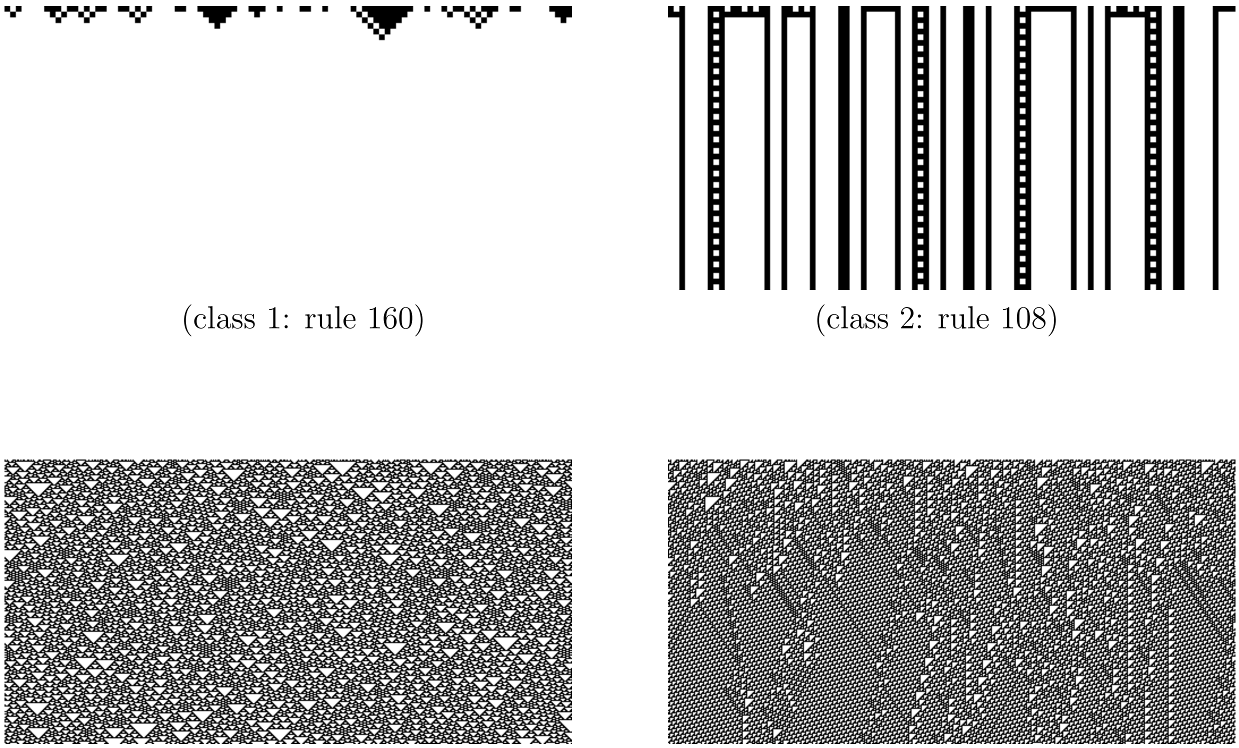
(W4) Localized structures with complex interactions emerge.

Figure 6 shows examples of typical space-time diagrams in each class. Wolfram conjectured that class (W4) cellular automata are computationally universal. In addition to rule 110 also elementary CA 54 is in class (W4).

1.6 Finite configurations

Let $s \in S$ be an arbitrary state. The *s-support* of configuration $c \in S^{\mathbb{Z}^d}$ is the set

$$supp_s(c) = \{\vec{n} \in \mathbb{Z}^d \mid c(\vec{n}) \neq s\}$$



(class 1: rule 160)

(class 2: rule 108)

(class 3: rule 126)

(class 4: rule 110)

Figure 6: Space-time diagrams of sample cellular automata from each of the four Wolfram classes.

of cells not in state s. Configuration c is called *s-finite* if $supp_s(c)$ is a finite set, that is, all but a finite number of cells are in state s. Let us denote

$$\mathcal{F}_s(d,S) = \{c \in S^{\mathbb{Z}^d} \mid \ c \text{ is } s\text{-finite } \}.$$

Note that $\mathcal{F}_s(d,S)$ is countably infinite while $S^{\mathbb{Z}^d}$ is uncountable.

Sometimes one state $q \in Q$ is identified as the *quiescent* state of the CA. The quiescent state q must satisfy

$$f(q,q,\ldots,q) = q,$$

that is, a cell whose neighbors are all quiescent becomes quiescent. If a quiescent state q is identified and fixed then the q-support of c is called simply the *support* of c and denoted by $supp(c)$. Moreover, q-finite configurations are called simply *finite*, and the set of finite configurations in $S^{\mathbb{Z}^d}$ is denoted by $\mathcal{F}(d,S)$, or simply by $\mathcal{F}$ when d and S are clear from the context. The configuration in which every cell is in state q is called the *quiescent configuration.*

Clearly, if c is s-finite then $G(c)$ is t-finite where $t = f(s, s \dots, s)$. In particular, in the presence of quiescent state q, finite configurations are mapped into finite configurations. In this case we denote by

$$G_F : \mathcal{F} \longrightarrow \mathcal{F}$$

the restriction of G on finite configurations.

In Example 1 (*xor* CA), we can name state 0 quiescent, in which case the space-time diagram in Figure 3 depicts a time-evolution according to G_F. In *Game-of-life* (Section 1.2) the white square (no life) is taken as the quiescent state.

1.7 Periodic configurations

Let $\vec{r} \in \mathbb{Z}^d$. Assuming a fixed and known state set S, the *translation* $\tau_{\vec{r}}$ determined by $\vec{r}$ is the global transition function of the CA whose neighborhood contains only $\vec{r}$ and whose local rule is the identity function. In other words,

$$\tau_{\vec{r}} : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$$

maps $c \mapsto c'$ where $c'(\vec{n}) = c(\vec{n} + \vec{r})$ for all $\vec{n} \in S^{\mathbb{Z}^d}$. It is obvious that for all $\vec{r}, \vec{s} \in \mathbb{Z}^d$ and $k \in \mathbb{Z}$ we have

$$\begin{aligned} \tau_{\vec{r}} \circ \tau_{\vec{s}} &= \tau_{\vec{r}+\vec{s}}, \text{ and} \\ \tau_{\vec{r}}^k &= \tau_{k\vec{r}}. \end{aligned} \tag{3}$$

For each dimension $i = 1, 2, \dots, d$ we call the translation by one cell down in dimension i a *shift* and denote it by σ_i. More precisely, if we denote the i'th coordinate unit vector

$$\vec{e_i} = (0, \dots, 0, 1, 0, \dots 0),$$

then $\sigma_i = \tau_{\vec{e_i}}$. It follows from (3) that every translation is a composition of shifts. In the one-dimensional case the only shift σ_1 is called the left shift and we denote it simply by σ.

The following proposition states an elementary but important property of cellular automata, based on the fact that all cells use the same local update rule:

Proposition 3 *Let G be an arbitrary CA function and τ a translation. Functions G and τ commute, i.e., $G \circ \tau = \tau \circ G$:*

$$\begin{tikzcd} S^{\mathbb{Z}^d} \arrow[r, "G"] \arrow[d, "\tau"] & S^{\mathbb{Z}^d} \arrow[d, "\tau"] \\ S^{\mathbb{Z}^d} \arrow[r, "G"] & S^{\mathbb{Z}^d} \end{tikzcd}$$

Proof. Let τ be the translation determined by $\vec{r} \in \mathbb{Z}^d$, and let G be the transition function of CA $A = (d, S, N, f)$ where N is as in (1). For arbitrary $c \in S^{\mathbb{Z}^d}$ and $\vec{n} \in \mathbb{Z}^d$ we have

$$\begin{aligned}
\tau(G(c))(\vec{n}) &= G(c)(\vec{n}+\vec{r}) \\
&= f[c(\vec{n}+\vec{r}+\vec{n}_1), c(\vec{n}+\vec{r}+\vec{n}_2), \ldots, c(\vec{n}+\vec{r}+\vec{n}_m)] \\
&= f[\tau(c)(\vec{n}+\vec{n}_1), \tau(c)(\vec{n}+\vec{n}_2), \ldots, \tau(c)(\vec{n}+\vec{n}_m)] \\
&= G(\tau(c))(\vec{n}),
\end{aligned}$$

so $\tau(G(c)) = G(\tau(c))$ and, furthermore, $G \circ \tau = \tau \circ G$. □

A configuration $c \in S^{\mathbb{Z}^d}$ is called $\vec{r}$*-periodic* if

$$c(\vec{n}) = c(\vec{n}+\vec{r}) \text{ for all } \vec{n} \in \mathbb{Z}^d.$$

Another way to say this is $c = \tau_{\vec{r}}(c)$, i.e., c is invariant under the translation by $\vec{r}$. A configuration is called *spatially periodic* if it is $\vec{r}$-periodic for some $\vec{r} \neq \vec{0}$.

A d-dimensional configuration is *totally periodic* if it is $\vec{r}_i$-periodic for some linearly independent $\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_d \in \mathbb{Z}^d$. It follows easily that a totally periodic configuration is σ_i^k-periodic for some $k \in \mathbb{Z}_+$ and all $i = 1, 2, \ldots, d$. In other words, a totally periodic configuration consists of a hypercubic pattern(D, p) that is repeated periodically in each of the d-dimensions of the space. Let us denote by $\mathcal{P}(d, S)$ the set of totally periodic elements of $S^{\mathbb{Z}^d}$, or if d and S are clear from the context we may simply denote $\mathcal{P}$ instead of $\mathcal{P}(d, S)$. Set $\mathcal{P}$ is countably infinite.

In the one-dimensional case there is no difference between spatial periodicity and total periodicity. In two- and higher dimensional spaces there is a difference. Figure 7(a) shows a two-dimensional configuration (infinite horizontal stripe) that is $\vec{e}_1$-periodic but not totally periodic. Figure 7(b) shows a totally periodic configuration (infinite checker board).

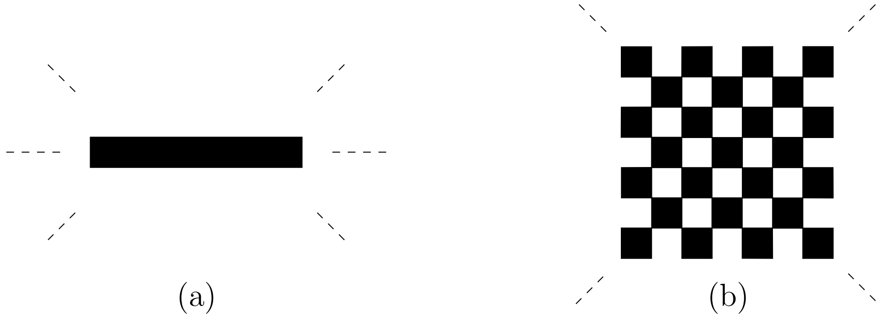


Figure 7: (a) A spatially periodic configuration that is not totally periodic, and (b) a totally periodic configuration.

Let G be a CA function and suppose configuration c is $\vec{r}$-periodic. According to Proposition 3

$$\tau_{\vec{r}}(G(c)) = G(\tau_{\vec{r}}(c)) = G(c)$$

so also $G(c)$ is $\vec{r}$-periodic. In particular, if c is totally periodic then also $G(c)$ is totally periodic. We denote by

$$G_P : \mathcal{P} \longrightarrow \mathcal{P}$$

the restriction of G on totally periodic configurations.

Finite configurations and periodic configurations are used in effective simulations of cellular automata on computers. Periodic configurations are often referred to as the *periodic boundary conditions* on a finite cellular array. For example, in the case $d = 2$ this is equivalent to running the CA on a torus that is obtained by "gluing" together the opposite sides of a rectangle. One should, however, keep in mind that the behavior of a CA can be quite different on finite, periodic and general configurations, so experiments done with periodic boundary conditions may sometimes be misleading.

One final remark: Periodicity of a configuration defined in this section refers to spatial periodicity. This should not be confused with temporal periodicity of a configuration defined at the end of Section 1.3, that is, the property that the configuration repeats itself under the CA evolution.

1.8 Compactness principle

Topology plays an important role in the theory of cellular automata. The configuration space $S^{\mathbb{Z}^d}$ can be given a compact topology under which all CA functions G are continuous. We delay the detailed discussion of this. Instead we prove two statements that capture essential features of the topological approach.

Consider an infinite sequence $c_1, c_2, \ldots$ of configurations, each $c_i \in S^{\mathbb{Z}^d}$. We say that the sequence *converges* and $c \in S^{\mathbb{Z}^d}$ is its *limit* if for every $\vec{n} \in \mathbb{Z}^d$ there exists some $k \in \mathbb{Z}_+$ such that $c_i(\vec{n}) = c(\vec{n})$ for all $i \geq k$. In other words: if we look at an arbitrary cell and browse through a converging sequence $c_1, c_2, \ldots$ then from some moment on we always see the same state. It is obvious that if a limit exists it is unique, and we denote this limit by

$$\lim_{i \to \infty} c_i.$$

A *subsequence* of $c_1, c_2, \ldots$ is another sequence $c_{i_1}, c_{i_2}, \ldots$ where $i_1 < i_2 < \ldots$. A subsequence is hence obtained by picking infinitely many elements of the sequence, preserving their relative order. Obviously every subsequence of a converging sequence also converges and has the same limit.

The first proposition states the compactness of the configuration space:

Proposition 4 *Every sequence of configurations has a converging subsequence.*

Proof. Let $c_1, c_2, \ldots$ be an arbitrary sequence, $c_i \in S^{\mathbb{Z}^d}$. Let $\vec{r}_1, \vec{r}_2, \ldots$ be an enumeration of elements of $\mathbb{Z}^d$. In the following we find a subsequence $c_{i_1}, c_{i_2}, \ldots$ such that

$$\forall k \in \mathbb{Z}_+ \; \exists s \in S \; \forall j \geq k : c_{i_j}(\vec{r}_k) = s. \tag{4}$$

(For all $k \in \mathbb{Z}_+$, all configurations c_{i_j} with $j \geq k$ have the same state in position $\vec{r}_k$.) Then clearly the subsequence converges.

In the following we choose inductively the indices $i_1 < i_2 < i_3 < \ldots$. For every $k \in \mathbb{Z}_k$ we denote by

$$I_k = \{i \in \mathbb{Z}_+ \mid \forall j = 1, 2, \ldots, k \;\; c_i(\vec{r}_j) = c_{i_k}(\vec{r}_j)\}$$

the set of indices of those configurations that coincide with the k'th choice c_{i_k} in the first k cells $\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_k$. The idea is to always choose the next index $i_{k+1} > i_k$ from I_k in such a way that I_{k+1} is infinite.

More precisely, we initialize $i_0 = 0$ and $I_0 = \mathbb{Z}_+$. For all $k = 0, 1, 2, \ldots$ we let i_{k+1} be the smallest element of I_k that is greater than i_k and that makes the next set I_{k+1} infinite. If I_k is infinite then such a choice can be made: there are only finitely many different possibilities for the state in cell $\vec{r}_{k+1}$, so there are infinitely many $i \in I_k$ with identical $c_i(\vec{r}_{k+1})$.

With these choices sets I_k are all infinite, and we obtain a sequence of indices $i_1 < i_2 < i_3 < \ldots$ such that (4) holds. □

Note: The proof is essentially the same as the proof of weak Kőnig's lemma which states that an infinite binary tree contains an infinite path. The proof did not require the axiom of choice. (The same result could also be briefly proved using Tychonoff's theorem, but that is equivalent to the axiom of choice.)

Our next proposition states a continuity property of CA functions:

Proposition 5 *Let G be a CA function and $c_1, c_2, \ldots$ a converging sequence of configurations. Then also the sequence $G(c_1), G(c_2), \ldots$ converges and*

$$\lim_{i\to\infty} G(c_i) = G(c)$$

where

$$c = \lim_{i\to\infty} c_i.$$

Proof. Let G be the transition function of $A = (d, S, N, f)$ where N is as in (1). Let $\vec{n} \in \mathbb{Z}^d$ be arbitrary. Because $c = \lim_{i\to\infty} c_i$ we have that for every $j = 1, 2, \ldots, m$ there exists $k_j \in \mathbb{Z}_+$ such that

$$c_i(\vec{n} + \vec{n_j}) = c(\vec{n} + \vec{n_j}) \text{ for all } i \geq k_j.$$

Let $k = \max\{k_1, k_2, \ldots, k_m\}$. Then if $i \geq k$ we have

$$\begin{aligned} G(c_i)(\vec{n}) &= f[c_i(\vec{n} + \vec{n_1}), c_i(\vec{n} + \vec{n_2}), \ldots, c_i(\vec{n} + \vec{n_m})] \\ &= f[c(\vec{n} + \vec{n_1}), c(\vec{n} + \vec{n_2}), \ldots, c(\vec{n} + \vec{n_m})] \\ &= G(c)(\vec{n}). \end{aligned}$$

Because $\vec{n} \in \mathbb{Z}^d$ was arbitrary, we have that $G(c_1), G(c_2), \ldots$ converges to $G(c)$. □

Our last proposition states that the sets of finite and totally periodic configurations are dense:

Proposition 6 *Let $c \in S^{\mathbb{Z}^d}$ and $s \in S$. There exist sequences*

(a) $c_1, c_2, \dots$ of s-finite configurations $c_i \in \mathcal{F}_s(d,S)$, and

(b) $p_1, p_2, \dots$ of totally periodic configurations $p_i \in \mathcal{P}(d,S)$

such that $c = \lim_{i\to\infty} c_i = \lim_{i\to\infty} p_i$.

Proof. Let $\vec{r}_1, \vec{r}_2, \dots$ an enumeration of $\mathbb{Z}^d$, and define, for every $i, j \in \mathbb{Z}_+$,

$$c_i(\vec{r}_j) = \begin{cases} c(\vec{r}_j), & \text{if } j \le i, \\ s, & \text{if } j > i, \end{cases}$$

It is clear that $c = \lim_{i\to\infty} c_i$.

For the analogous claim concerning totally periodic configurations, denote $D_i = \{-i, \dots, i\}^d$ and set, for every $i \in \mathbb{Z}_+$, configuration p_i to be the totally periodic configuration that coincides with c in D_i and has period $2i+1$ in each coordinate direction, that is, p_i satisfies $\sigma_j^{2i+1}(p_i) = p_i$ for all $j = 1, 2, \dots, d$. □

2 Injectivity and surjectivity properties

2.1 Basic facts

Let $g : A \longrightarrow B$ be a function. Recall the following notation and terminology: For any $K \subseteq A$ we denote the image of K by

$$g(K) = \{g(k) \mid k \in K\},$$

and for any $L \subseteq B$ we denote the pre-image of L by

$$g^{-1}(L) = \{a \in A \mid g(a) \in L\}.$$

For $b \in B$ the set

$$g^{-1}(b) = \{a \in A \mid g(a) = b\}$$

is the set of pre-images of element b. Function $g : A \longrightarrow B$ is called

- *injective* or *one-to-one* if every element of B has at most one pre-image:

$$|g^{-1}(b)| \le 1 \text{ for all } b \in B,$$

- *surjective* or *onto* if every element of B has at least one pre-image:

$$|g^{-1}(b)| \geq 1 \text{ for all } b \in B,$$

- *bijective* if it is both injective and surjective, that is, every element of B has exactly one pre-image:

$$|g^{-1}(b)| = 1 \text{ for all } b \in B,$$

A CA is called injective, surjective or bijective if its transition function G is injective, surjective or bijective, respectively. In this section we investigate these properties and their relation to injectivity, surjectivity and bijectivity of the restricted functions G_F and G_P. In particular, the goal is to prove the implications shown in Figure 8 for one-dimensional CA and in Figure 9 for two- and higher dimensional CA.

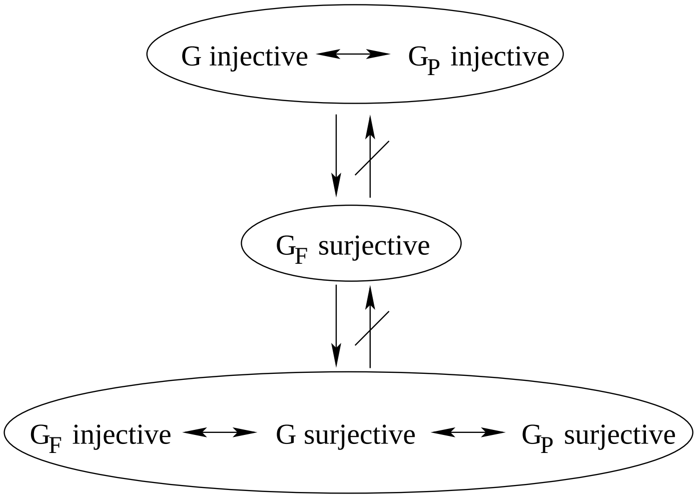


Figure 8: Implications between injectivity and surjectivity properties in one-dimensional CA.

We can make the following initial observations:

Proposition 7 *For any CA function G holds:*

(a) If G is injective then also G_F and G_P are injective.

(b) If G_F or G_P is surjective then also G is surjective.

(c) If G_P is injective then G_P is surjective.

Proof. (a) is obvious from the definition of injectivity. For (b), suppose that every q-finite configuration has a pre-image. Let $c \in S^{\mathbb{Z}^d}$ be an arbitrary configuration. In the following

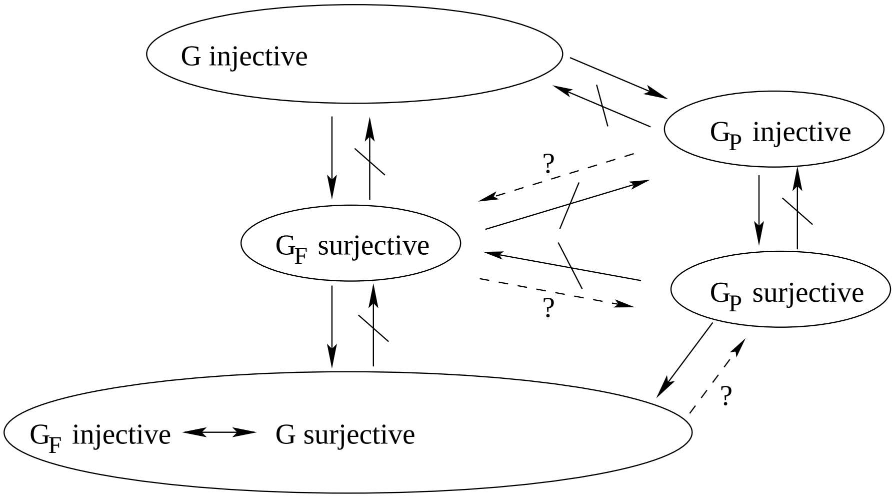


Figure 9: Implications between injectivity and surjectivity properties in two- and higher dimensional CA.

we use the compactness property to show that c has a pre-image. By Proposition 6 there exists a sequence of finite configurations $c_1, c_2, \ldots$ that converges to c. By the hypothesis, for every $i \in \mathbb{Z}_+$ there exists a configuration e_i such that $G(e_i) = c_i$. According to Proposition 4, the sequence $e_1, e_2, \ldots$ has a converging subsequence $e_{i_1}, e_{i_2}, \ldots$. Let e be the limit of the subsequence. By Proposition 5 the sequence $G(e_{i_1}), G(e_{i_2}), \ldots$ converges to $G(\lim_{j\to\infty} e_{i_j}) = G(e)$. On the other hand,

$$\lim_{j\to\infty} G(e_{i_j}) = \lim_{j\to\infty} c_{i_j} = \lim_{i\to\infty} c_i = c$$

so we have $c = G(e)$.

The proof of (b) for G_P is analogous.

Consider then claim (c). Let c be a totally periodic configuration. It means that there are positive integers $k_1, k_2, \ldots, k_d$ such that

$$\sigma_i^{k_i}(c) = c, \text{ for all } i = 1, 2, \ldots, d. \tag{4}$$

Let us denote by $\mathcal{K}$ the set of configurations that satisfy (4) for the fixed numbers $k_1, k_2, \ldots, k_d$. Set $\mathcal{K}$ is finite (containing $|S|^{k_1 k_2 \ldots k_d}$ elements), $\mathcal{K} \subseteq \mathcal{P}$, and by Proposition 3 we have $G(\mathcal{K}) \subseteq \mathcal{K}$. If G_P is injective, so is G restricted to $\mathcal{K}$. It follows then from the finiteness of $\mathcal{K}$ that $G(\mathcal{K}) = \mathcal{K}$. It means that every element of $\mathcal{K}$, including c, has a periodic pre-image. We have shown that G_P is surjective. □

Corollary 8 *Every injective CA is surjective, so injectivity is equivalent to bijectivity.*

Proof. G injective $\Longrightarrow G_P$ injective $\Longrightarrow G_P$ surjective $\Longrightarrow G$ surjective. □

2.2 Reversible CA

A cellular automaton function G is called *reversible* if it is bijective and the inverse function G^{-1} is also a CA function. A cellular automaton A is called reversible if its global transition function G is reversible. Then the CA computing G^{-1} is called the *inverse automaton* of A, and we denote it by A^{-1}. We know from Proposition 2 that the inverse automaton is unique up to adding dummy neighbors and ordering of the neighbors. The inverse of A retraces the orbits of A backwards in time.

Example 4. Let $d = 1$, $S = \{1, 2, 3\}$, $N = (0, 1)$, and the value $f(a, b)$ is given by the following table:

$a \backslash b$	1	2	3
1	1	1	2
2	2	2	1
3	3	3	3

(States 1 and 2 get swapped if the right neighbor is 3.) This CA G is reversible. In fact, it is its own inverse, that is, $G^2 = id$. Notice how the inverse rule gives $c(n)$ based on $G(c)(n)$ and $G(c)(n+1)$ even though $c(n)$ does not influence $G(c)(n+1)$ in any way in the forward direction. □

Every reversible CA has to be bijective by definition. The following proposition shows that the converse is also true. Note that this is not obvious: one could expect that in some bijective CA a cell might need to look at cells arbitrarily far away in order to determine its previous state. That, however, never happens:

Proposition 9 *Every bijective CA is reversible.*

Proof. Suppose CA function G is bijective but not reversible. Let $\vec{r}_1, \vec{r}_2, \ldots$ be an enumeration of elements of $\mathbb{Z}^d$. For every $i \in \mathbb{Z}_+$ there exist configurations c_i and e_i such that

$$c_i(\vec{0}) \neq e_i(\vec{0}), \text{ and} \tag{5}$$

$$G(c_i)(\vec{r_j}) = G(e_i)(\vec{r_j}), \text{ for all } j \leq i. \tag{6}$$

Namely, if such c_i and e_i did not exist then $G(c)(\vec{r_1}), \ldots, G(c)(\vec{r_i})$ would uniquely determine $c(\vec{0})$ in every configuration c. Then a CA with neighborhood $\{\vec{r}_1, \ldots, \vec{r}_i\}$ would define a CA function H satisfying $G \circ H = id$, the identity function. Since G is bijective, this would mean that $H = G^{-1}$, contradicting the initial hypotheses that G is not reversible.

A parallel version (homework assignment) of Proposition 4 implies that there exist indices $i_1 < i_2 < i_3 < \ldots$ such that subsequences

$$c_{i_1}, c_{i_2}, \ldots$$
$$e_{i_1}, e_{i_2}, \ldots$$

both converge. Let

$$\begin{aligned} c &= \lim_{j\to\infty} c_{i_j}, \\ e &= \lim_{j\to\infty} e_{i_j} \end{aligned}$$

be their limits. According to (5) we have $c_{i_j}(\vec{0}) \neq e_{i_j}(\vec{0})$ for every j, so $c(\vec{0}) \neq e(\vec{0})$. This means that $c \neq e$.

On the other hand, it follows from the continuity property (Proposition 5) that sequences

$$\begin{array}{l} G(c_{i_1}), G(c_{i_2}), \ldots \\ G(e_{i_1}), G(e_{i_2}), \ldots \end{array}$$

converge, and

$$\begin{aligned} \lim_{j\to\infty} G(c_{i_j}) &= G(c), \\ \lim_{j\to\infty} G(e_{i_j}) &= G(e). \end{aligned}$$

But it follows from (6) that the limits must be same, so $G(c) = G(e)$, which contradicts the bijectivity of G.

□

By Proposition 9 and Corollary 8, injectivity, bijectivity and reversibility are equivalent concepts on cellular automata.

Corollary 10 *If G is injective then G_F is surjective.*

Proof. If G is injective then it is reversible by Proposition 9 and Corollary 8. If q is the quiescent state of G then it is also quiescent in the inverse CA, that is, the local rule g of the inverse CA maps $(q, q, \ldots, q) \mapsto q$. Hence, if c is a finite configuration, also $e = G^{-1}(c)$ is finite and $G(e) = c$, so c has a finite pre-image. □

2.3 Balance in surjective CA

A configuration c is a *Garden-of-Eden* configuration (GOE) if it has no pre-images, i.e. if $G^{-1}(c)$ is empty. A CA has Garden-of-Eden configurations if and only if the CA is not surjective.

Example 5. Consider the elementary CA number 110 from Example 3. Among eight possible neighborhood patterns there are three that are mapped to state 0 and five that are mapped to state 1. Let us demonstrate how this imbalance automatically implies that there are Garden-of-Eden configurations.

Let k be a positive integer, and consider a configuration c in which

$$c(3) = c(6) = \ldots = c(3k) = 0.$$

See Figure 10 for an illustration. There are

$$2^{2(k-1)} = 4^{k-1}$$

possible choices for the missing states between 0's in c (shown as "*" in Figure 10).

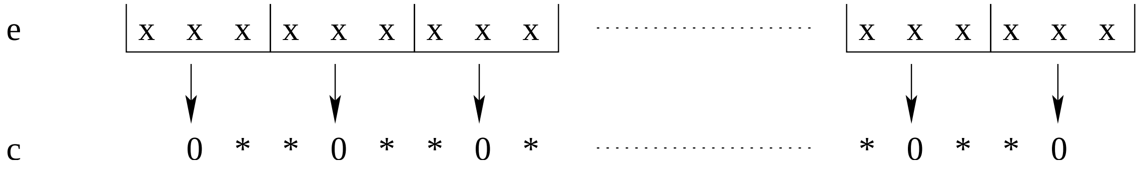


Figure 10: Illustration of the configuration c and its pre-image e in Example 5.

In a pre-image e of c the three state segments $e(i-1), e(i), e(i+1)$ are mapped into state 0 by the local rule f, for every $i = 1, 2, \ldots, k$. Since $|f^{-1}(0)| = 3$ there are exactly 3^k choices of these segments.

If k is sufficiently large then $3^k < 4^{k-1}$. This means that that some choice of c does not have a corresponding pre-image e. Therefore the CA is not surjective.

Alternatively, one could show the non-surjectivity of rule 110 by directly verifying that any configuration containing pattern 01010 is a Garden-of-Eden.

□

In this section we generalize the previous example. We need first to define the concept of a (finite) pattern. A *pattern*

$$p = (D, g)$$

is a partial configuration where $D \subseteq \mathbb{Z}^d$ is the *domain* of p and $g : D \longrightarrow S$ is a mapping assigning a state to each cell in the domain. Pattern p is *finite* if D is a finite set. Note that configurations are patterns whose domain is the entire space $\mathbb{Z}^d$. If τ is a translation of $\mathbb{Z}^d$ then $\tau(p)$ is the pattern $p' = (D', g')$ where $D' = \tau(D)$ and $g = \tau \circ g'$. We then say that p and p' are *translated copies* of each other.

If $p_1 = (D_1, g_1)$ and $p_2 = (D_2, g_2)$ are two patterns we say that p_1 is a *subpattern* of p_2 if $D_1 \subseteq D_2$ and $g_1(\vec{n}) = g_2(\vec{n})$ for all $\vec{n} \in D_1$. We say that p_2 *contains a copy* of p_1 if some translated copy of p_1 is a subpattern of p_2. Patterns p_1 and p_2 are *disjoint* if $D_1 \cap D_2 = \emptyset$.

Let G be CA function specified by CA $A = (d, S, N, f)$, where N is given by (1). Let $p = (D, g)$ be a pattern, and let $D' \subseteq \mathbb{Z}^d$ a domain such that $N(D') \subseteq D$, that is, all neighbors of all cells of D' are in D. An application of the local rule f on pattern p determines new states for all cells in domain D'. We obtain a pattern $p' = (D', g')$ where for all $\vec{n} \in D'$

$$g'(\vec{n}) = f[g(\vec{n} + \vec{n}_1), g(\vec{n} + \vec{n}_2), \ldots, g(\vec{n} + \vec{n}_m)].$$

The mapping $p \mapsto p'$ will be denoted by $G^{(D \to D')}$, or simply by G when the domains D and D' are clear from the context and there is no risk of confusion. Note that the global transition function of the CA is $G^{(\mathbb{Z}^d \to \mathbb{Z}^d)}$.

A finite pattern without a pre-image is called an *orphan*. In other words, pattern $p' = (D', g')$ is an orphan if $G^{(D \to D')}(p) \neq p'$ for all $p = (D, g)$ with domain $D = N(D')$. Clearly any configuration that contains a copy of an orphan is a Garden-of-Eden configuration. Also the converse is true, as stated in the next proposition. The proof is similar to Proposition 7(b).

Proposition 11 *Every Garden-of-Eden configuration has a subpattern that is an orphan. Hence, a cellular automaton is non-surjective if and only if there exists an orphan.*

Proof. Let $c \in S^{\mathbb{Z}^d}$ be a Garden-of-Eden configuration, and suppose that none of its subpatterns is an orphan. Let $\vec{r}_1, \vec{r}_2, \ldots$ be an enumeration of elements of $\mathbb{Z}^d$ and denote, for every $j \in \mathbb{Z}_+$,

$$D_j = \{\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_j\}.$$

Since the subpattern of c with domain D_j is not an orphan, there exists a configuration $c_j \in S^{\mathbb{Z}^d}$ such that $G(c_j)$ agrees with c in domain D_j. This implies that the sequence $G(c_1), G(c_2), \ldots$ converges to c. By compactness (Proposition 4) the sequence $c_1, c_2, \ldots$ has a converging subsequence $c_{i_1}, c_{i_2}, \ldots$, with some limit $e \in S^{\mathbb{Z}^d}$. By the continuity of G (Proposition 5) the sequence $G(c_{i_1}), G(c_{i_2}), \ldots$ converges to $G(e)$. On the other hand,

$$\lim_{j \to \infty} G(c_{i_j}) = \lim_{i \to \infty} G(c_i) = c.$$

So we have $c = G(e)$, which means that c is not a GOE. □

The following lemma is a technical result that will be needed in this and the next section:

Lemma 12 *For all $d, n, s, r \in \mathbb{Z}_+$ there exists $k \in \mathbb{Z}_+$ such that*

$$\left(s^{n^d} - 1\right)^{k^d} \; < \; s^{(kn-2r)^d}.$$

Proof. A homework assignment. □

The d-dimensional *hypercube* of size n^d determined by corner $(k_1, k_2, \ldots, k_d) \in \mathbb{Z}^d$ is the finite domain

$$D = \{(x_1, x_2, \ldots, x_d) \in \mathbb{Z}^d \mid k_i \leq x_i < k_i + n \text{ for all } i = 1, 2, \ldots, d\ \}.$$

Patterns with hypercubic domains will be extensively used in proofs below. Now we are ready to state and prove the balance property of surjective CA:

Proposition 13 *Let $A = (d, S, N, f)$ be a surjective CA, and let $D, D' \subseteq \mathbb{Z}^d$ be finite domains such that $N(D') \subseteq D$. Then for every pattern $p' = (D', g')$ the number of patterns $p = (D, g)$ such that*

$$G^{(D \to D')}(p) = p'$$

is $s^{|D|-|D'|}$ where $s = |S|$ is the number of states.

Proof. Suppose there exists p' such that the number of pre-image patterns p in domain D is $t \neq s^{|D|-|D'|}$. Let us first show that we can assume that the domains D and D' are hypercubes, with D' centered inside D.

Let $E, E' \subseteq \mathbb{Z}^d$ be arbitrary finite domains such that $D \subseteq E$, $D' \subseteq E'$ and $N(E') \subseteq E$. In particular, for sufficiently large n and r we can choose E and E' to be cocentric hypercubes of size n^d and $(n-2r)^d$, respectively. There are $s^{|E'|-|D'|}$ patterns in domain E' that have p' as a subpattern (one can choose arbitrary states in the cells in $E' \setminus D'$, a shaded region in Figure 11), and they have $t \cdot s^{|E|-|D|}$ pre-image patterns in domain E (obtained from the t pre-images of p' by choosing arbitrary states in the remaining cells in $E \setminus D$, also shaded in Figure 11). If each pattern in domain E' would have $s^{|E|-|E'|}$ pre-images in domain E, we would have

$$s^{|E'|-|D'|} \cdot s^{|E|-|E'|} = t \cdot s^{|E|-|D|}.$$

This implies $t = s^{|D|-|D'|}$, a contradiction. We conclude that there is a pattern in domain E' that has an "imbalanced" number of pre-images in domain E.

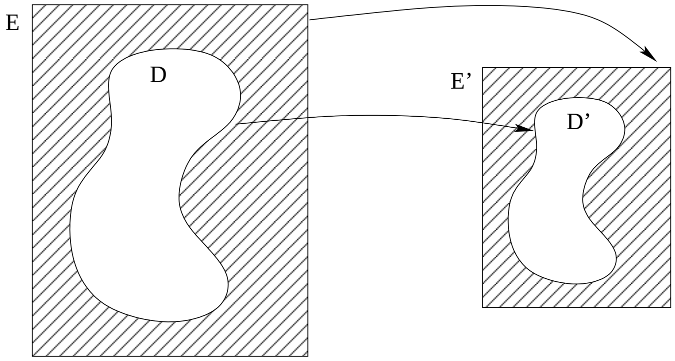


Figure 11: Domains D, D', E and E'.

In the following we hence assume that D and D' are cocentric hypercubes of size n^d and $(n-2r)^d$, respectively. We also assume that the number t of pre-images of $p' = (D', g')$ in domain D satisfies

$$t < s^{|D|-|D'|}.$$

Namely, the total number of patterns in domains D and D' are $s^{|D|}$ and $s^{|D'|}$, respectively, so if every pattern in domain D' would have at least $s^{|D|-|D'|}$ pre-images in domain D, then every pattern would necessarily have exactly $s^{|D|-|D'|}$ pre-images, contradicting the assumption that p' has an imbalanced number of pre-images.

The main part of the proof that follows is similar to Example 5. Let $k \in \mathbb{Z}_+$ be arbitrary. Consider a domain C that is a hypercube of size $(kn)^d$, partitioned into k^d non-overlapping hypercubes of size n^d. See Figure 12 for an illustration. Let C' be the hypercube of size $(nk-2r)^d$ centered inside C, and let us denote by $\mathcal{A}$ the set of finite patterns with domain C' such that each of the k^d shaded domains in Figure 12 contain a copy of pattern p'. The states in the $|C'| - k^d|D'|$ non-shaded cells of C' may be chosen freely, so

$$|\mathcal{A}| = s^{|C'|-k^d|D'|}.$$

Let $\mathcal{B}$ be the set of finite patterns of domain C whose image under $G^{(C \to C')}$ is in $\mathcal{A}$. They are exactly the patterns that contain one of the t pre-images of p' in each of the k^d sub-hypercubes, so

$$|\mathcal{B}| = t^{k^d}.$$

Let us show that if k is sufficiently large then

$$t^{k^d} < s^{|C'|-k^d|D'|}, \tag{7}$$

that is, $|\mathcal{A}| > |\mathcal{B}|$. This proves that there is a pattern in $\mathcal{A}$ without a pre-image, contradicting the surjectivity of G.

We have

$$\begin{array}{rcl} |D| & = & n^d \text{ and} \\ |C'| & = & (kn-2r)^d, \end{array}$$

so according to Lemma 12 there is k such that

$$\left(s^{|D|}-1\right)^{k^d} \ < \ s^{|C'|}.$$

Then

$$\begin{array}{rcl} t^{k^d} & \leq & \left(s^{|D|-|D'|}-1\right)^{k^d} \\ & \leq & \left(s^{|D|-|D'|}-s^{-|D'|}\right)^{k^d} \\ & = & s^{-k^d|D'|}\left(s^{|D|}-1\right)^{k^d} \\ & < & s^{|C'|-k^d|D'|}, \end{array}$$

which proves (7).

□

As a special case we get that the local rule table of a surjective cellular automaton is balanced:

Corollary 14 *In surjective CA*

$$|f^{-1}(a)| = |S|^{m-1}$$

for all $a \in S$, where m is the size of the neighborhood and

$$f^{-1}(a) = \{(s_1, \ldots, s_m) \mid f(s_1, \ldots, s_m) = a\}.$$

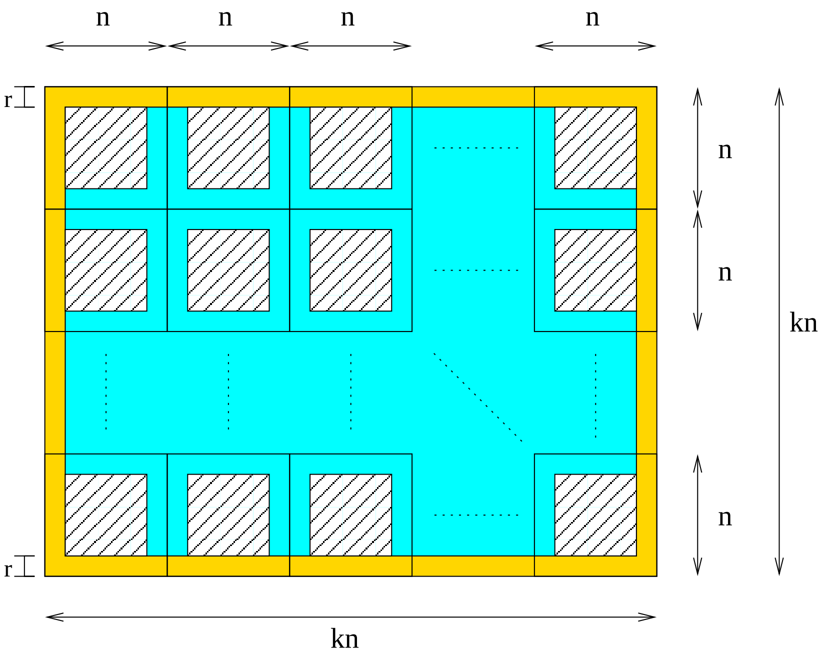


Figure 12: Illustration for the proofs of Propositions 13, 16 and 18.

Proof. Choose $D' = \{\vec{0}\}$ in the proposition. □

Example 6. Unlike the general balance property of Proposition 13, the balance condition of the local rule stated in Corollary 14 is not sufficient for surjectivity. Consider, for example, the elementary CA number 232. It is the *majority* CA: $f(a, b, c) = 1$ if and only if $a+b+c \geq 2$. Its rule table is balanced because $000, 001, 010, 100$ map to 0 and $111, 110, 101, 011$ map to 1.

However, the majority CA is not balanced on longer patterns and hence it is not surjective: Any pattern of length four that contains at most one state 1 is mapped to 00, so 00 has at least 5 pre-images of length four. Balanceness would require this number of pre-images to be 4.

□

2.4 Garden-of-Eden -theorem

One of the oldest results in cellular automata theory is the so-called Garden-of-Eden -theorem that states that there are Garden-of-Eden configurations if and only if there are different finite configurations with the same image. In other words: G is surjective if and only if G_F is injective. The two directions of the statement were proved by E.F.Moore in 1962 and J.Myhill in 1963.

A natural way to state the Garden-of-Eden -theorem without any reference to a quiescent state is in terms of pre-injectivity. Configurations c_1 and c_2 are called *asymptotic* if the set

$$\mathit{diff}(c_1, c_2) = \{\vec{n} \in \mathbb{Z}^d \mid c_1(\vec{n}) \neq c_2(\vec{n})\ \}$$

of positions where c_1 and c_2 differ is finite. Cellular automaton G is *pre-injective* if for any asymptotic c_1 and c_2 holds $c_1 \neq c_2 \Longrightarrow G(c_1) \neq G(c_2)$. Clearly all injective CA are pre-injective.

The following proposition shows that for pre-injectivity it is enough that the CA is one-to-one among c-asymptotic configurations, for any fixed configuration c. In particular – by choosing as c any q-finite configuration – we see that pre-injectivity is equivalent to injectivity of G_F.

Proposition 15 *Let $c \in S^{\mathbb{Z}^d}$ be arbitrary. Cellular automaton G is pre-injective if and only if it is injective in the domain*

$$\mathit{asymp}(c) = \{e \in S^{\mathbb{Z}^d} \mid\ c \textit{ and } e \textit{ are asymptotic }\}.$$

Proof. It is clear that pre-injectivity implies injectivity in domain $\mathit{asymp}(c)$. For the converse direction, suppose that G is injective in $\mathit{asymp}(c)$, and let c_1 and c_2 be two asymptotic configurations, $c_1 \neq c_2$. Assume that we could have $G(c_1) = G(c_2)$.

Let $N \subseteq \mathbb{Z}^d$ be the set of the elements of the neighborhood vector. We may assume $\vec{0} \in N$: if not, we simply add $\vec{0}$ as a dummy neighbor. Denote by

$$A = \mathit{diff}(c_1, c_2) - N$$

the set of cells that have a neighbor in $\mathit{diff}(c_1, c_2)$, and by

$$B = A + N = \mathit{diff}(c_1, c_2) + N - N$$

the neighborhood of A. Clearly A and B are finite sets.

Consider the configurations e_1 and e_2 where $e_i(\vec{n}) = c_i(\vec{n})$ for all $\vec{n} \in B$, and $e_i(\vec{n}) = c(\vec{n})$ for $\vec{n} \notin B$. Because $c_1 \neq c_2$ and $\mathit{diff}(c_1, c_2) \subseteq B$, we also have $e_1 \neq e_2$. But $G(e_1) = G(e_2)$ because

- for $\vec{n} \in A$ the neighborhood of $\vec{n}$ is inside B, so $G(e_1)(\vec{n}) = G(c_1)(\vec{n}) = G(c_2)(\vec{n}) = G(e_2)(\vec{n})$, and
- for $\vec{n} \notin A$, the neighborhood of $\vec{n}$ does not contain elements of $\mathit{diff}(c_1, c_2)$, so configurations e_1 and e_2 are identical in the neighborhood of cell $\vec{n}$. Hence $G(e_1)(\vec{n}) = G(e_2)(\vec{n})$.

Configurations e_1 and e_2 are asymptotic to c, which contradicts the injectivity of G in the set $\mathit{asymp}(c)$. □

Example 7. As in the previous section, we start by illustrating the proof of the Garden-of-Eden -theorem using elementary rule 110. We know from Example 5 that rule 110 is not surjective. In fact, finite pattern 01010 has no pre-image. Let us demonstrate that there must exist different 0-finite configurations c and e such that $G(c) = G(e)$.

Let $k \in \mathbb{Z}_+$ be arbitrary. Consider finite configurations c whose supports are included in a fixed segment of length $5k - 2$, see Figure 13. There are

$$2^{5k-2} = 32^k/4$$

such configurations. The support of $G(c)$ is included in a segment of length $5k$. Partition this segment in k subsegments of length 5. We know that pattern 010101 cannot appear anywhere in $G(c)$, so there are at most $2^5 - 1 = 31$ different patterns that can appear in the length 5 subsegments. Hence there are at most 31^k possible configurations $G(c)$. For all sufficiently large values of k we have

$$32^k/4 > 31^k$$

so there must be two finite configurations with the same image, and G is not pre-injective.

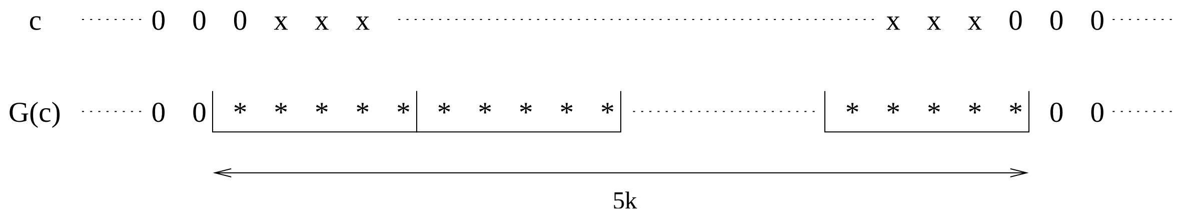


Figure 13: Illustration of Example 7.

□

Proposition 16 *If G is not surjective then G is not pre-injective.*

Proof. Suppose G is not surjective and let $q \in S$ be arbitrary. We show that there are two different q-finite configurations c_1 and c_2 such that $G(c_1) = G(c_2)$. Let $f(q, q, \ldots q) = t$, and let $s = |S|$ be the number of states. Choose $r \in \mathbb{Z}_+$ sufficiently big so that a radius-r CA defines G.

By Proposition 11 there exists a finite pattern p such that any configuration that contains p is Garden-of-Eden. We can pad p with copies of state t so that the domain of p becomes a size n^d hypercube.

Let $k \in \mathbb{Z}_+$ be arbitrary, and consider a size $(kn)^d$ hypercube C. Exactly as in the proof of Proposition 13 we partition C into k^d non-overlapping hypercubes of size n^d, as illustrated in Figure 12. Let C' be the hypercube of size $(nk - 2r)^d$ centered inside C. Let

$$K = \{c \in S^{\mathbb{Z}^d} \mid supp_q(c) \subseteq C' \ \}$$

be the set of q-finite configurations whose non-q states are inside hypercube C'. There are $s^{|C'|}$ elements in K.

The t-support of $G(c)$ for every $c \in K$ is inside C. Moreover, $G(c)$ cannot contain pattern p in any of the k^d sub-hypercubes of size n^d. It means that there are at most

$$\left(s^{n^d} - 1\right)^{k^d}$$

possible configurations $G(c)$. But according to Lemma 12 for some k

$$\left(s^{n^d} - 1\right)^{k^d} \; < \; s^{(kn-2r)^d} = s^{|C'|} = |K|,$$

so there are $c_1, c_2 \in K$ such that $c_1 \neq c_2$ while $G(c_1) = G(c_2)$. □

Corollary 17 *If G_F is injective then G is surjective.*

Proof. If G_F is injective then G is pre-injective by Proposition 15, and then by Proposition 16 it is surjective. □

Note that the implication chain

$$G \text{ injective} \implies G \text{ pre-injective} \implies G \text{ surjective}$$

provides a second proof of Corollary 8 that uses asymptotic pairs of configurations instead of periodic configurations.

Next we turn to the other direction of the Garden-of-Eden -theorem. Again, we start with a one-dimensional example that indicates the proof idea.

Example 8. Consider again rule 110. The 0-finite configurations

$$c_1 = \ldots 000011010000 \ldots$$
$$c_2 = \ldots 000010110000 \ldots$$

have the same image. Let us demonstrate how this implies that rule 110 is not surjective. (Of course we already know this fact form the prior examples.)

Extract patterns $p_1 = 00110100$ and $p_2 = 00101100$ of length eight from c_1 and c_2, respectively. Both patterns are mapped into the same pattern 111110 of length six. Moreover, p_1 and p_2 have a boundary of width 2 on both sides where they are identical with each other. Since rule 110 uses radius-1 neighborhood, one can replace in any configuration c pattern p_1 by p_2 or vice versa without affecting $G(c)$.

Let $k \in \mathbb{Z}_+$, and consider a segment of $8k$ cells. It consists of k segments of length 8. Any pattern of length $8k-2$ that has a pre-image of length $8k$ also has a pre-image where none of the k subsegments of length 8 contains pattern p_1. Namely, all such p_1 can be replaced by

p_2. This means that at most $(2^8-1)^k = 255^k$ patterns of length $8k-2$ can have pre-images. On the other hand there are $2^{8k-2} = 256^k/4$ such patterns, and for large values of k

$$256^k/4 > 255^k,$$

so some patterns do not have a pre-image. □

Proposition 18 *If G is not pre-injective then G has Garden-of-Eden configurations.*

Proof. Suppose c_1 and c_2 are asymptotic, $c_1 \neq c_2$ but $G(c_1) = G(c_2)$. Let r be sufficiently large so that G is defined by a radius-$\frac{r}{2}$ cellular automaton. Choose n sufficiently large so that there is a size $(n-2r)^d$ hypercube D' containing all cells where c_1 and c_2 differ, that is,

$$\mathit{diff}(c_1, c_2) \subseteq D'.$$

Let D be the size n^d hypercube around D' that is cocentric with D', and let $p_1 = (D, g_1)$ and $p_2 = (D, g_2)$ be the subpatterns of c_1 and c_2 with domain D, respectively.

In any configuration c that has subpattern p_1 we can replace that subpattern p_1 by p_2 without affecting $G(c)$. Indeed, those cells $\vec{n}$ whose neighborhood does not contain elements of $\mathit{diff}(c_1, c_2)$ do not see any change in their neighborhood. Those cells $\vec{n}$ whose neighborhood contains elements of $\mathit{diff}(c_1, c_2)$ are within distance $\frac{r}{2}$ of D', so their neighborhood is entirely inside D. Therefore

$$G(c)(\vec{n}) = G(c_1)(\vec{n}) = G(c_2)(\vec{n}) = G(c')(\vec{n})$$

where c' is the configuration obtained from c by replacing subpattern p_1 by p_2.

As in the proofs of Propositions 13 and 16, let $k \in \mathbb{Z}_+$ and let C be a hypercube of size $(kn)^d$, consisting of k^d non-overlapping sub-hypercubes of size n^d. Let C' be the hypercube of size $(kn-2r)^d$ centered inside C, see Figure 12. If G is surjective then every pattern with domain C' has a pre-image in domain C, and based on the discussion above, it has a pre-image where none of the k^d sub-hypercubes of size n^d contains a copy of p_1. But there are only

$$\left(s^{|D|}-1\right)^{k^d} = \left(s^{n^d}-1\right)^{k^d}$$

such patterns in domain C, while there are

$$s^{(kn-2r)^d}$$

patterns in domain C'. It follows from Lemma 12 that some pattern does not have a pre-image.

□

Corollary 19 *If G is surjective then G_F is injective.* □

Example 9. In Game-of-Life configurations c_1 where all cells are dead and c_2 that has exactly one living cell have the same image. Hence Game-of-Life is not surjective. By Proposition 11 there exist orphans. Interestingly, no very small orphans for Game-of-Life are known. Currently the smallest known example has a domain of size 88, see Figure 14.

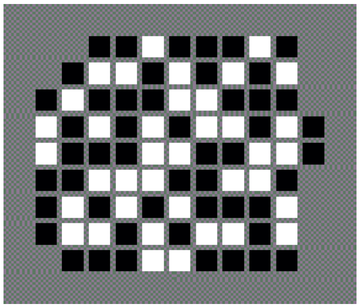


Figure 14: Smallest known orphan in Game-of-Life (due to Steven Eker in 2017). Black cells are living. It is known that there are no orphans with 6×6 square domain.

2.5 One-dimensional case

In this section we concentrate on one-dimensional cellular automata. The balanceness of surjective CA has the following interesting corollary that is valid in the one-dimensional case only:

Proposition 20 *For every one-dimensional surjective CA there is a constant n such that every configuration has at most n pre-images.*

Proof. Let G be a one-dimensional surjective CA function. Let r be sufficiently large so that G is defined by a radius-r cellular automaton and let $s = |S|$ be the number of states. In the following we prove that every configuration has at most s^{2r} different pre-images.

Suppose the contrary: there is a configuration c with $s^{2r} + 1$ different pre-images

$$e_1, e_2, \ldots, e_{s^{2r}+1}.$$

For some sufficiently large number $k > r$, all pairs of pre-images e_i and e_j contain a difference inside the interval

$$D = \{-k, -k+1, \ldots, k-1, k\}.$$

In other words, for all i, j with $1 \leq i < j \leq s^{2r} + 1$ there is $l \in D$ such that $e_i(l) \neq e_j(l)$. But this contradicts Proposition 13 since we now have a pattern with domain

$$D' = \{-k+r, -k+r+1, \ldots, k-r\}$$

that has

$$s^{2r} + 1 > s^{|D|-|D'|}$$

different pre-images in domain D. □

Corollary 21 *Let G be a one-dimensional surjective CA function. If $c \in S^{\mathbb{Z}}$ is not periodic then $G(c)$ is not periodic either. In particular, G_P is surjective.*

Proof. Suppose $G(c)$ is periodic, so that $\sigma^n(G(c)) = G(c)$ for some $n \in \mathbb{Z}_+$. Then for every $i \in \mathbb{Z}$

$$G(\sigma^{in}(c)) = \sigma^{in}(G(c)) = G(c),$$

so $\sigma^{in}(c)$ is a pre-image of $G(c)$. By Proposition 20 configuration $G(c)$ has a finite number of pre-images so

$$\sigma^{i_1 n}(c) = \sigma^{i_2 n}(c)$$

for some $i_1 < i_2$. But then c is periodic with period $(i_2 - i_1)n$. It follows that all pre-images of periodic configurations are periodic. □

Proposition 20 and Corollary 21 do not hold for two-dimensional surjective cellular automata as shown by the following example:

Example 10. Consider the two-dimensional *xor* CA with radius-$\frac{1}{2}$ neighborhood: $d = 2$, $S = \{0, 1\}$,

$$N = [(0,0), (0,1), (1,0), (1,1)]$$

and the local rule $f : \{0,1\}^4 \longrightarrow \{0,1\}$ is

$$f(a,b,c,d) = \begin{cases} 0, & \text{if } a+b+c+d \text{ is even,} \\ 1, & \text{if } a+b+c+d \text{ is odd.} \end{cases}$$

This CA is surjective: otherwise there would be two different finite configurations $c, e \in \{0,1\}^{\mathbb{Z}^2}$ with the same image $G(c) = G(e)$, see Proposition 16. If $(x,y) \in \mathbb{Z}^2$ is a cell where $c(x,y) \neq e(x,y)$ then it follows from the local rule of the CA that $c(x',y') \neq e(x',y')$ for $(x',y') = (x+1,y), (x,y+1)$ or $(x+1,y+1)$. In any case $x' + y' > x + y$, which implies that the set of cells where c and e differ cannot be a finite set.

Even though the CA is surjective the (totally periodic) quiescent configuration c_0 in which all cells are in state 0 has uncountably many pre-images. For example, any configuration c that consists of horizontal stripes, i.e.

$$c(i,k) = c(j,k)$$

for all $i, j, k \in \mathbb{Z}$, is a pre-image of c_0. Many of these pre-images are not totally periodic. However, the second part of Corollary 21 is not refuted by this example since G_P is surjective. It is not known whether the surjectivity of G always implies the surjectivity of G_P in two- and higher dimensional CA. □

The following proposition states a converse of Proposition 20. This statement is valid in any dimension:

Proposition 22 *If G is a non-surjective CA then there is a totally periodic configuration $c \in \mathcal{P}$ that has infinitely (even uncountably) many pre-images.*

Proof. Homework.

We can, in fact, be more specific about the structure of the pre-images in surjective one-dimensional CA. Let us call two one-dimensional configurations $c, e \in S^{\mathbb{Z}}$ *positively asymptotic* (*negatively asymptotic*) if $c(i) = e(i)$ for all sufficiently large (all sufficiently small, respectively) $i \in \mathbb{Z}$. Let us call c and e *positively n-separated* (*negatively n-separated*) if for all sufficiently large (sufficiently small, respectively) $i \in \mathbb{Z}$ there is $j \in \{i, i+1, i+2, \ldots, i+n-1\}$ such that $c(j) \neq e(j)$. We say that c and e are *positively separated* (*negatively separated*) if they are positively n-separated (negatively n-separated, respectively) for some $n \in \mathbb{Z}_+$. Configurations c and e are *totally n-separated* if for all $i \in \mathbb{Z}$ there is $j \in \{i, i+1, i+2, \ldots, i+n-1\}$ such that $c(j) \neq e(j)$, and c and e are *totally separated* if they are totally n-separated for some n. Clearly c and e are totally separated if and only if they are both positively and negatively separated (but the separation parameter n may be different).

Also the following terminology related to one-dimensional neighborhoods will be used: Number m is a neighborhood *range* of a CA function G if G is defined by a CA whose neighborhood consists of m consecutive integers. In particular, a radius-$\frac{1}{2}$ CA has neighborhood range 2, and a radius-r CA has range $2r + 1$, for any $r \in \mathbb{Z}_+$.

Proposition 23 *Let G be a one-dimensional surjective CA function with neighborhood range m, and let $c, e \in S^{\mathbb{Z}}$ be such that $c \neq e$ and $G(c) = G(e)$. Then exactly one of the following three conditions is true:*

(i) c and e are negatively asymptotic and positively $(m-1)$-separated,

(ii) c and e are positively asymptotic and negatively $(m-1)$-separated, or

(iii) c and e are both positively and negatively $(m-1)$-separated.

Proof. Conditions (i)–(iii) are pairwise exclusive so it is enough to show that at least one of them holds for c and e.

Let us first show that if $c(n) \neq e(n)$ then there cannot be segments of length $m-1$ on both sides of n where c and e agree. Suppose the contrary: there exist $k_1 < n$ and $k_2 > n$ such that $c(i) = e(i)$ for all i in the intervals $k_1-(m-1) < i \leq k_1$ and $k_2 \leq i < k_2+(m-1)$. Then we can replace in configuration c the states in cells $k_1 \ldots k_2$ by the states of the corresponding cells in configuration e without affecting $G(c)$. This contradicts Proposition 18 since we obtain a configuration c' that only differs from c in a finite number of cells while $G(c') = G(c)$.

Now it easily follows that c and e must be either positively asymptotic or positively $(m-1)$-separated. Otherwise there would be a segment of length $(m-1)$ where c and e agree, a position n to the right of this segment where $c(n) \neq e(n)$, and another segment of length $(m-1)$ to the right of n where c and e again agree.

A symmetric reasoning shows that c and e must be negatively asymptotic or negatively $(m-1)$-separated. Now it remains to notice that c and e cannot be both positively and negatively asymptotic as then c and e would be in contradiction to Proposition 18.

□

The following example shows that all conditions (i)–(iii) of the previous proposition are indeed possible. It also shows that the surjectivity of G does not imply the surjectivity of G_F, which in turn does not imply the injectivity of G.

Example 11. Consider the one-dimensional CA with three states $0, 1$, and 2 and the radius-$\frac{1}{2}$ neighborhood, where the local rule is

$$f(a,b) = \begin{cases} 2, & \text{if } a = 2, \\ 0, & \text{if } a \neq 2 \text{ and } a+b \text{ is even, and} \\ 1, & \text{if } a \neq 2 \text{ and } a+b \text{ is odd.} \end{cases}$$

The rule keeps state 2 unchanged, while other states are changed as in the *xor* CA where state 2 as the right neighbor behaves as 0. Notice that the local rule is *left permutive*: For every fixed b the mapping $a \mapsto f(a,b)$ is a permutation of the state set $S = \{0, 1, 2\}$.

Let us first verify that the CA is surjective: Consider two configurations c and e that are different but have the same image. Let $n \in \mathbb{Z}$ be any position where $c(n) \neq e(n)$. Because $G(c)(n) = G(e)(n)$ it follows directly from the left permutativity of f that $c(n+1) \neq e(n+1)$. This means that $c(i) \neq e(i)$ for all $i \geq n$, so according to Garden-of-Eden -theorem (Proposition 16) the CA is surjective.

Configurations

$$\begin{gathered} \ldots 000020000 \ldots \\ \ldots 000021111 \ldots \end{gathered}$$

are negatively asymptotic and positively 1-separated. Clearly they have the same image. Configurations

$$\begin{gathered} \ldots 00000000 \ldots \\ \ldots 11111111 \ldots \end{gathered}$$

are totally 1-separated and have the same image.

The examples above mean also that G is not injective. But it is surjective on finite configurations if state 2 is taken as the quiescent state. Indeed: It follows from the surjectivity that the CA is one-to-one on finite configurations. Since the 2-support of c and $G(c)$ are always identical, and since there are finitely many configurations with any given finite support, it follows that every finite configuration has a pre-image with the same support. Hence this example shows that the surjectivity of G_F does not imply the injectivity of G.

On the other hand, if state 0 is taken as the quiescent state then G_F is not surjective: The configuration

$$\ldots 000010000 \ldots$$

with single state 1 has only non-finite pre-images. So we also see that the surjectivity of G does not imply the surjectivity of G_F. (The *xor* CA would have provided another example of this.)

□

Proposition 7, Corollaries 17, 10, 19 and 21, and the previous Example 11 contain all results but one summarized in Figure 8. The last remaining implication is proved next:

Proposition 24 *Let G be a one-dimensional CA function. If G_P is injective then G is injective.*

Proof. Let m be a neighborhood range for G. Suppose G is not injective, so there are $c, e \in S^{\mathbb{Z}}$ such that $c \neq e$ and $G(c) = G(e)$. Since G_P is injective it follows from Propositions 7 that G is surjective. According to Proposition 23 c and e are positively or negatively $(m-1)$-separated. The two alternatives are symmetric, so let us assume without loss of generality that c and e are positively $(m-1)$-separated.

There are only finitely many different patterns of length $m-1$ in c and e, so there exist arbitrarily large positive numbers k_1 and k_2 such that, in both c and e, the patterns of length $m-1$ starting in positions k_1 and k_2 are identical. More precisely, there are $k_2 \geq k_1 + m$ such that

$$c(k_1 + i) = c(k_2 + i) \text{ and } e(k_1 + i) = e(k_2 + i) \text{ for all } 0 \leq i < m - 1.$$

Because c and e are positively $(m-1)$-separated, we take k_1 sufficiently large so that $c(k_1 + i) \neq e(k_1 + i)$ for some i in the interval $0 \leq i < m - 1$.

Consider the periodic configurations c_p and e_p that are invariant under the translation by $k_2 - k_1$ cells and agree with c and e, respectively, in cells $k_1, k_1 + 1, \ldots k_2 - 1$. More precisely, for $k_1 \leq i < k_2$ and $n \in \mathbb{Z}$

$$c_p(i + n(k_2 - k_1)) = c(i) \text{ and } e_p(i + n(k_2 - k_1)) = e(i)$$

Then for every $j \in \mathbb{Z}$ there is some i in the interval $k_1 \leq i < k_2$ such that the length m segments in c_p and e_p starting in position j are the same as the length m segments in c and e starting in position i, respectively. Because m is a neighborhood range for G and $G(c) = G(e)$ it follows that $G(c_p) = G(e_p)$. On the other hand, $c_p \neq e_p$ so that G_P is not injective. We reached a contradiction. □

2.6 De Bruijn -graphs

Let us continue with one-dimensional cellular automata. In this section we introduce a new way to represent the local rule as a labeled directed graph. A directed (multi)graph has a finite set V of *vertices* or *nodes*, and another finite set E of directed *edges*. Functions $t : E \longrightarrow V$ and $h : E \longrightarrow V$ give the *tail* $t(e)$ and the *head* $h(e)$ of edge $e \in E$. We say that edge e is from vertex $t(e)$ into vertex $h(e)$. This formalism allows multiple edges between nodes. If $t(e) = h(e)$ then edge e is called a *loop*. We often draw directed graphs as diagrams where edges $e \in E$ are drawn as arrows pointing from node $t(e)$ into node $h(e)$, see e.g. Figure 15 for examples of such diagrams. Frequently we do not write in the diagram the names of the edges and only show the corresponding arrows. The *outdegree* of vertex v is the number of edges whose tail is v, and its *indegree* is the number of edges whose head is v.

A *path* (of length k) is a sequence $e_1, e_2, \ldots, e_k$ of edges where $h(e_i) = t(e_{i+1})$ for all $i = 1, 2, \ldots, k-1$, that is, paths "follow the arrows" in the diagram representation of the graph. A two-way infinite path is a sequence $p : \mathbb{Z} \longrightarrow E$ such that for every $i \in \mathbb{Z}$ we have $h(p(i)) = t(p(i+1))$.

An (edge) labeled directed graph is a directed graph together with a labeling function $l : E \longrightarrow \Sigma$ which assigns each edge a symbol from a finite set Σ of labels. The label of a finite or infinite path is the sequence of elements of Σ obtained by reading the labels of its edges. For instance, in the two-way infinite case, the label of path $p : \mathbb{Z} \longrightarrow E$ is the sequence $l_p \in \Sigma^{\mathbb{Z}}$ where $l_p(i) = l(p(i))$ for all $i \in \mathbb{Z}$.

Let m be a positive integer and let S be a finite set. The *de Bruijn* graph of width m over alphabet S is the directed graph with

$$\begin{aligned} V &= S^{m-1}, \\ E &= S^m, \\ t(s_1 s_2 \ldots s_m) &= s_1 s_2 \ldots s_{m-1}, \text{ and} \\ h(s_1 s_2 \ldots s_m) &= s_2 s_3 \ldots s_m. \end{aligned}$$

In other words, there is an edge from node su to node ut for all $s, t \in S$ and $u \in S^{m-2}$. This overlap property means that for every $c \in S^{\mathbb{Z}}$ there is a two-way infinite path $p : \mathbb{Z} \longrightarrow E$ such that

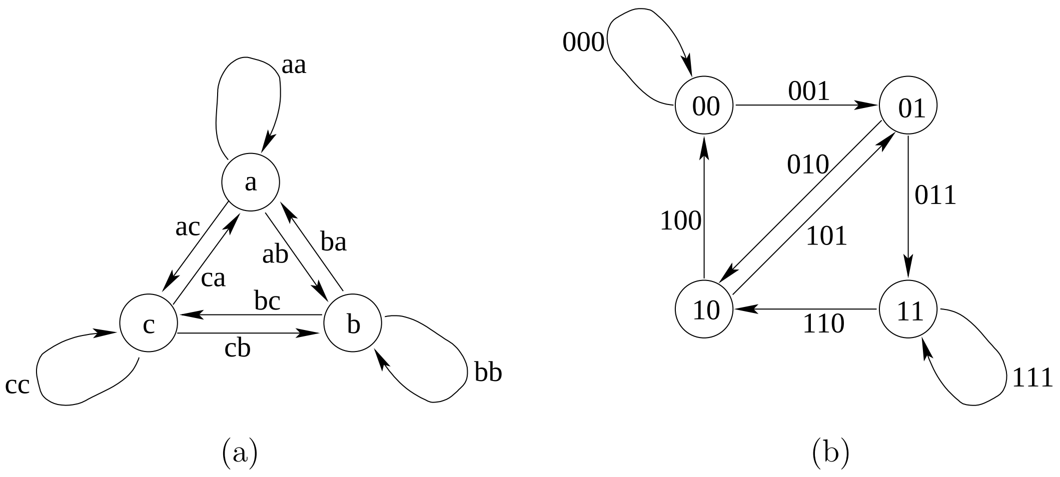
$$p(i) = c(i)c(i+1)\ldots c(i+m-1) \text{ for all } i \in \mathbb{Z}.$$

Path p is obtained by sliding a window of width m over c. The edges along p are the views through the sliding window.

The correspondence $c \leftrightarrow p$ is bijective: c is obtained from path p by reading the first components of the edges along the path. Let us denote the configuration c that corresponds to path p by c_p.

Example 12. Figure 15(a) shows the de Bruijn graph of width $m = 2$ over three letter alphabet $S = \{a, b, c\}$, while Figure 15(b) shows the de Bruijn graph of width $m = 3$ over two letter alphabet $S = \{0, 1\}$. In general, the de Bruijn graph of width m over k symbols has k^{m-1} vertices and k^m edges. The graph is balanced in the sense that each vertex has indegree k and outdegree k. □

Let m be a neighborhood range of a one-dimensional cellular automaton function G, which means that G is specified by a CA A whose neighborhood is a segment of m consecutive integers and the local rule is a function $f : S^m \longrightarrow S$. The labeled de Bruijn graph associated with A is the de Bruijn graph of width m over alphabet S in which each edge $e \in S^m$ is labeled by $f(e) \in S$. In other words, for all $s_1, s_2, \ldots, s_m$ the graph has an edge labeled by $f(s_1, s_2, \ldots, s_m)$ from vertex $s_1 s_2 \ldots s_{m-1}$ into vertex $s_2 s_3 \ldots s_m$. This labeled graph is called the *de Bruijn representation* of CA A. It contains full information about the local



(a)

(b)

Figure 15: De Bruijn graphs of width (a) $m = 2$ over $\{a, b, c\}$, and (b) $m = 3$ over $\{0, 1\}$.

rule of the CA. Note, however, that since the position of the CA neighborhood is not given, CA functions $G \circ \sigma^k$ are represented by the same de Bruijn graph for all $k \in \mathbb{Z}$. Fortunately this is not a problem since we use the de Bruijn representations to study properties such as injectivity and surjectivity that are not affected by translations.

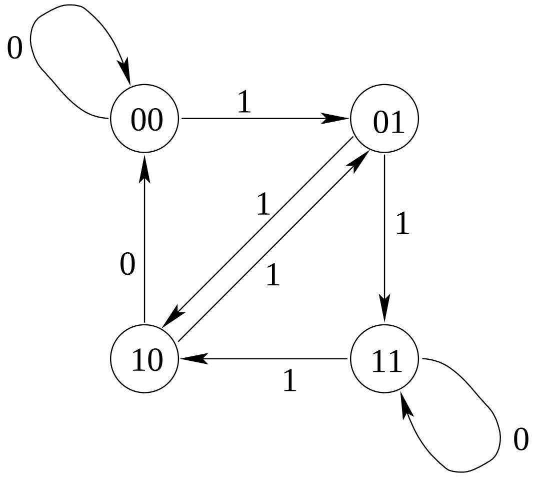


Figure 16: The de Bruijn representation of rule 110.

Example 13. Figure 16 shows the labeled de Bruijn graph of rule 110. □

Two-way infinite paths p in the de Bruijn representations provide two elements of $S^{\mathbb{Z}}$: We have c_p, the sequence obtained by reading the first symbol of the names of the edges along p, and we have f_p, the sequence obtained by reading the labels of the edges. Because the labels are the outputs of the local update rule of the CA, it is clear that f_p is (possibly translated) $G(c_p)$:

$$\sigma^k(f_p) = G(c_p) \text{ for some } k \in \mathbb{Z}.$$

The amount k of the translation depends on the positioning of the neighborhood in the CA: The neighborhood is $\{k, k+1, \ldots, k+m-1\}$.

Now we can interpret previously discussed CA properties in the de Bruijn representation:

- The CA is injective if and only if different two-way infinite paths have always different labels.
- The CA is surjective if and only if for every $c \in S^{\mathbb{Z}}$ there is a path labeled by c. By the Garden-of-Eden -theorem this is equivalent to saying that the graph does not have a *diamond*: A diamond consists of two different finite paths with identical labels that begin in the same vertex and end in the same vertex.
- An orphan is a word over alphabet S that is not a label of any path.

In the following we see how de Bruijn representations provide practical algorithms to determine if a given one-dimensional CA is injective or surjective. But let us first see how we can find the orphans of a given non-surjective CA. As mentioned above an orphan is a word that is not a label of any path. To find such a word we use the *subset construction* to make a *deterministic finite automaton* that recognizes all orphans.

Let us first briefly review some concepts of automata theory. A labeled digraph is also called a finite *semiautomaton*. Vertices are in this context called the *states*, the labels are *letters*, and the edges are *transitions*. If we also specify two subsets $I \subseteq V$ and $F \subseteq V$ of vertices we obtain a *finite automaton*. Elements of I and F are called *initial* and *final states*, respectively. The automaton is used to accept words over the finite labeling alphabet Σ. A *word* is a finite sequence of elements of Σ. Number of symbols in the sequence is the *length* of the word. The word of length zero is the *empty word*, and we denote it by ε. The set of all words is

$$\Sigma^* = \bigcup_{k=0}^{\infty} \Sigma^k$$

where Σ^k is the set of words of length k. A set $L \subseteq \Sigma^*$ of words is called a *language*. Finite automaton *accepts* a word $w \in \Sigma^*$ if there exists a path labeled by w that starts in some initial state and ends in a final state. The language *recognized* by a finite automaton is the set of all words that it accepts. Languages that are recognized by finite automata are called *regular*. We call two finite automata *equivalent* if they recognize the same language.

Using the automata theoretic terminology we note that the orphans of a cellular automaton are precisely the words that are not accepted by the automaton that we get from the de Bruijn graph by making all states initial and final.

A finite automaton is called *deterministic* if there is only one initial state, and for each state $v \in V$ and letter $a \in \Sigma$ there is at most one transition with label a from state v. Since there is now only one possible continuation from every state with each letter, it is clear that for every input word $w \in \Sigma^*$ there is at most one path that starts in the initial state. Word w is accepted if and only if the last state of this path is a final state. A deterministic finite automaton is *complete* if there is a (unique) transition from every state with every input letter. It is easy to make any deterministic finite automaton complete by adding a new state (which is not final) and making all missing transitions into this *sink* state. Clearly exactly the same words are accepted as before.

The power set construction is a way to convert an arbitrary finite automaton into an equivalent deterministic and complete automaton. The power set automaton has state set 2^V, that is, all subsets of V are states. For any $X \subseteq V$ and $a \in \Sigma$ the transition from X with input letter a is made into the state

$$\{v \in V| \text{ for some } x \in X \text{ there is an edge } x \to v \text{ with label } a \ \}.$$

One easily sees that in the power set automaton the last state of the path that starts at state $X \subseteq V$ and is labeled by word w consists of all those states $v \in V$ such that there is a path from some element of X into v labeled by w in the original automaton. In particular, if we make I the initial state, and make every set $X \subseteq V$ such that $X \cap F \neq \emptyset$ a final state of the power set automaton, then exactly the same words are accepted that were accepted in the original automaton. Moreover, if we swap the final states so that we instead make a state $X \subseteq V$ final iff $X \cap F = \emptyset$ then we have a deterministic automaton that accepts the complement language.

Let us perform the subset construction on the de Bruijn representation of a CA, where all states are considered initial and final. We obtain a complete deterministic finite automaton that accepts the words that are not orphans. Its initial state is S^{m-1} and all states except $\emptyset$ are final. Let's swap the final states, which means that $\emptyset$ becomes the only final state. Then we get an automaton that accepts exactly the orphans. We see:

Proposition 25 *The set of all orphans of a one-dimensional CA is a regular language.* □

Note: many states of the power set automaton may be unreachable from the initial state. Such states can be removed from the automaton without affecting the language it recognizes.

Example 14. The power set automaton of the de Bruijn representation of rule 110 is shown in Figure 17. The complete power set automaton contains $2^4 = 16$ states, but eight of them are not reachable from the initial state $\{00, 01, 10, 11\}$. In the construction of the power set automaton it is best to begin from the initial state S^{m-1} and add new states as they are reached. In this way only the reachable part ever gets constructed.

The label of the shortest path from the initial state $\{00, 01, 10, 11\}$ to the final state $\emptyset$ is 01010. It is hence the shortest orphan of rule 110. The shortest path can always be found using the breadth-first search in the graph. We can also make the observation that only words containing the pattern 010 can be orphans in rule 110, so any configuration which does not have isolated 1's has a pre-image. □

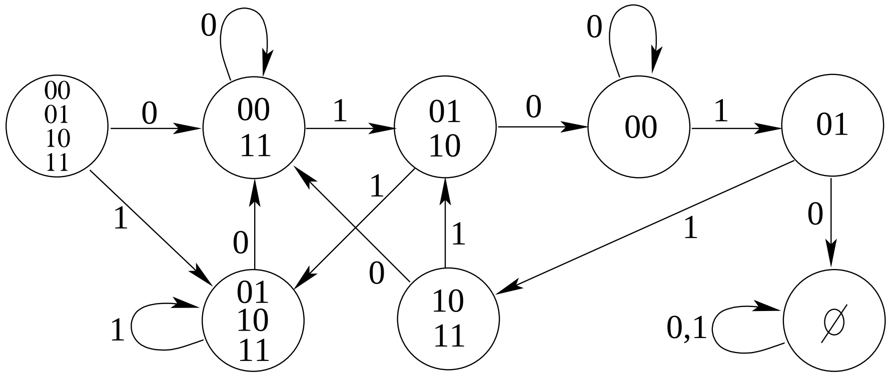


Figure 17: A deterministic automaton for the orphans in rule 110. State $\{00, 01, 10, 11\}$ is the initial state and state $\emptyset$ is the only final state.

Let us next move on to the problem of testing a one-dimensional CA function G for injectivity and surjectivity. Both these questions boil down to the question of which configurations have the same image under G. For this we use the following cartesian product construction.

From the de Bruijn representation of a CA we form a *pair graph* whose vertex set is $V \times V$ where $V = S^{m-1}$ is the vertex set of the de Bruijn representation. In the pair graph there is an edge (with label $a \in S$) from $(u_1, u_2) \in V \times V$ into $(v_1, v_2) \in V \times V$ if and only if in the de Bruijn graph there are edges with the same label a from u_1 to v_1 and from u_2 to v_2. See Figure 18 for the pair graph of rule 110.

Any two-way infinite path p in the pair graph corresponds to two paths in the original de Bruijn automaton, obtained by reading only the first or the second components of the pairs. Both paths have the same edge labels, so they correspond to two configurations of the CA with the same image. Let us denote these configurations by c_p^1 and c_p^2. The correspondence

$$p \leftrightarrow (c_p^1, c_p^2)$$

is a bijection between paths p and pairs of configurations that satisfy $G(c_p^1) = G(c_p^2)$.

Let us denote by

$$\Delta = \{(u, u) \mid u \in V\}$$

the set of diagonal vertices. Notice that the induced subgraph with vertex set Δ is an isomorphic copy of the de Bruijn automaton. In particular, there is a path between any two vertices in Δ, that is, it is strongly connected.

Any two-way infinite path p that only uses diagonal vertices has $c_p^1 = c_p^2$, so it does not provide two different configurations with the same image. Only paths that contain a vertex outside of Δ provide such configurations.

The following proposition states basic connections between the pair graph and the CA:

Proposition 26 *A one-dimensional CA A is*

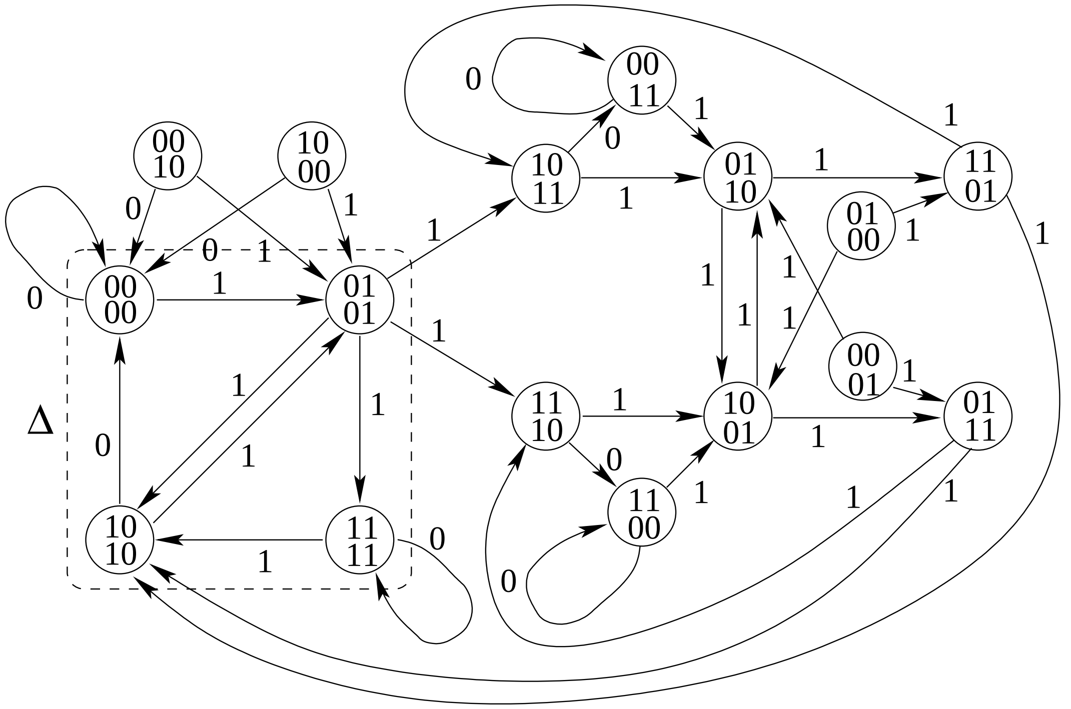


Figure 18: The complete pair graph of rule 110.

(i) not injective if and only if its pair graph has a cycle that contains a node outside of Δ*,*

(ii) not surjective if and only the pair graph has a cycle that contains a node of Δ *and a node outside of* Δ*.*

Proof. (i) If A is not injective then there are two different spatially periodic configurations c and e with the same image (Proposition 24). The path that corresponds to them in the pair graph is a cycle that contains a node outside of Δ. Conversely, if the pair graph has such a cycle then the corresponding configurations of the CA have the same image, i.e. the CA is not injective.

(ii) Let $q \in S$ be arbitrary. If A is not surjective then there are two different q-finite configurations c and e with the same image. The corresponding path in the pair graph consists of a loop inside Δ, followed by a cycle that goes outside of Δ and returns to the same loop inside Δ. Hence there is a cycle with nodes inside and outside of Δ. Conversely, if such a cycle exists, then the de Bruijn automaton has a diamond and the CA is not surjective. □

The size of the pair graph can be reduced by a factor of approximately two by observing a symmetry in the pair graph: For all $v, u \in V$ the states (u, v) and (v, u) are mirror images of each other, and can be merged. Also, since Δ is strongly connected and we are only interested in cycles that contain elements outside of Δ, we can merge all states of Δ into a single vertex (which we name Δ). The resulting directed graph is called the *reduced pair graph*. It follows from Proposition 26 that the CA is injective if and only if there is no cycle in its reduced pair graph, and the CA is surjective if and only if there is no cycle through node Δ. See Figure 19 for the reduced pair graph of rule 110. Standard depth-first algorithm of the reduced graph can be used to determine both these conditions in time that is linear in the size of the reduced pair graph.

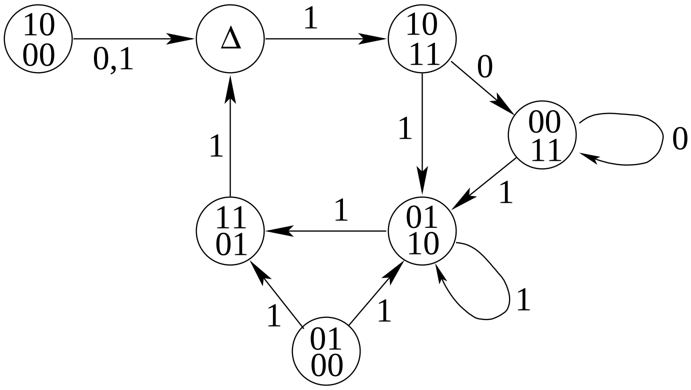


Figure 19: The reduced pair graph of rule 110.

Example 15. Consider the pair graph of rule 110, shown in Figure 18. One immediately sees from its cycles that the CA is not injective or surjective. The shortest path that begins and ends in node $(00, 00)$ and is not completely within Δ has length 6. The corresponding patterns (that have the same image 111110) are 00110100 and 00101100. They are the shortest pair of distinct patterns that begin and end in 00, and have the same image. □

Notice that Proposition 23 has a natural interpretation in the pair graph: Two-way infinite paths in the pair graph that

- after some time stay outside of Δ correspond to positively $(m-1)$-separated configurations,
- before some time stay outside of Δ correspond to negatively $(m-1)$-separated configurations, and
- always stay outside of Δ correspond to totally $(m-1)$-separated configurations

with the same image under G.

2.7 Two-dimensional CA and tilings

Proposition 7 and Corollaries 17, 10 and 19 prove all positive implications in Figure 9. For the negative implications that are common with the one-dimensional case we use the one-dimensional CA of Example 11. The d-dimensional version of this CA uses neighborhood $\{(0,0,\ldots,0),(1,0,0,\ldots,0)\}$ and the same local rule f. This means that the space consists of independently operating one-dimensional CA along the first dimension. It was shown in Example 11 that the one-dimensional version is surjective but not injective. The restriction G_F to finite configurations is surjective if 2 is taken as the quiescent state and not surjective if 0 is taken as the quiescent state. In the one-dimensional case injectivity and surjectivity of G are always equivalent to the injectivity and surjectivity, respectively, of G_P, the restriction of G to totally periodic configurations, so G_P in this example is surjective but not injective. Clearly the d-dimensional version has all these same properties, so this example provides all the negative implications shown in Figure 9 except for one. To complete the picture we need an example of a two-dimensional CA which is not reversible, while G_P is injective. For such an example, and for other forthcoming considerations on two-dimensional CA, we turn to plane tilings.

Tilings are a topic of another course *Tilings and Patterns*, so we refer to that class for more details and complete proofs of the results concerning tilings. However, we can easily present here the relevant results (without proofs), and see how they can be applied on cellular automata.

A *tile set* $\mathcal{T}=(T,N,R)$ consists of a finite set T whose elements are the *tiles*, a *neighborhood vector* N defined analogously to (1) as a vector of m distinct elements of $\mathbb{Z}^2$, and a *local matching rule* $R\subseteq T^m$ which gives a relation specifying which tilings are considered valid. *Tilings* are configurations $t\in T^{\mathbb{Z}^2}$. A tiling is valid at cell $\vec{n}\in\mathbb{Z}^2$ if and only if

$$[t(\vec{n}+\vec{n}_1),t(\vec{n}+\vec{n}_2),\ldots,t(\vec{n}+\vec{n}_m)]\in R,$$

that is, the neighborhood of $\vec{n}$ contains a matching combination of tiles. Tiling $t\in T^{\mathbb{Z}^2}$ is called *valid* if it is valid at all positions $\vec{n}\in\mathbb{Z}^2$, and we say that the tile set $\mathcal{T}$ then *admits* tiling t. Let $V(\mathcal{T})$ be the set of all valid tilings admitted by $\mathcal{T}$.

There is an apparent similarity in the definitions of tile sets and two-dimensional cellular automata. The only difference is that instead of a dynamic local rule f, tilings are based on a static matching relation R. In symbolic dynamics terminology, the fact that $V(\mathcal{T})$ is defined by forbidding a finite collection of patterns means that $V(\mathcal{T})$ is a two-dimensional subshift of finite type.

A fundamental property of cellular automata is that they commute with translations. The tiling counter part states the obvious fact that set $V(\mathcal{T})$ is invariant under translations. The second fundamental property is the continuity of CA functions. A tiling counter part of this fact states that the set of valid tilings is closed, that is, the limit of a converging sequence of valid tilings is also valid. Proofs are straightforward.

Proposition 27 *Let $\mathcal{T}=(T,N,R)$ be a tile set.*

(i) If t is a valid tiling and τ is a translation of the plane then $\tau \circ t$ is a valid tiling.

(ii) Suppose $t_1, t_2, \ldots$ is a converging sequence where for every $\vec{n} \in \mathbb{Z}^2$ there is k such that t_i is valid at cell $\vec{n}$ for all $i \geq k$. Then $\lim_{i\to\infty} t_i$ is a valid tiling.

□

A direct consequence of (ii) is that a tile set that can tile arbitrarily large squares, admits a tiling of the whole space:

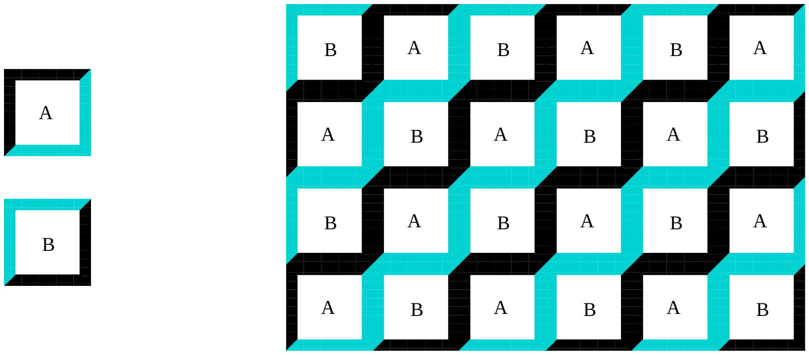
Corollary 28 *Suppose $\mathcal{T} = (T, N, R)$ is a tile set such that for every finite $D \subseteq \mathbb{Z}^2$ there is $t \in T^{\mathbb{Z}^2}$ that is valid at every $\vec{n} \in D$. Then $\mathcal{T}$ admits a valid tiling of the plane.*

Proof. Let $\vec{r}_1, \vec{r}_2, \ldots$ be an enumeration of elements of $\mathbb{Z}^2$, and for every $j \in \mathbb{Z}_+$ let

$$D_j = \{\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_j\}.$$

The hypothesis of the corollary states that there is $t_j \in T^{\mathbb{Z}^2}$ that is valid at every $\vec{n} \in D_j$. Let t be the limit of a converging subsequence of $t_1, t_2, \ldots$. It follows from Proposition 27(ii) that t is a valid tiling. □

A popular way to describe a tile set is to use *Wang tiles*. Wang tiles use the von Neumann neighborhood. The tiles are viewed as unit squares whose edges are colored, and the local matching rule is given in terms of these colors: A tiling is valid at cell $\vec{n} \in \mathbb{Z}^2$ iff each of the four edges of the tile in position $\vec{n}$ have the same color as the abutting edge in the adjacent tile.



(a) (b)

Figure 20: (a) Two Wang-tiles, and (b) part of a valid tiling.

Example 16. Consider the two Wang-tiles A and B shown in Figure 20(a). Since all four neighbors of tile A have to be copies of B, and vice versa, the only valid tilings are infinite checkerboards where A's and B's alternate, as shown in Figure 20(b). □

Let $\vec{r} \neq \vec{0}$. A tiling $t \in T^{\mathbb{Z}^2}$ is called *$\vec{r}$-periodic* if it is invariant under translation by $\vec{r}$, that is, if

$$t(\vec{n}) = t(\vec{n} + \vec{r}) \text{ for all } \vec{n} \in \mathbb{Z}^d.$$

A tiling is *totally periodic* if it is periodic with respect to two linearly independent vectors. A totally periodic tiling is always periodic with respect to σ_1^k and σ_2^k for some $k > 0$, where σ_1 and σ_2 are the horizontal and vertical shifts by one cell. The tiling in Figure 20(b) is totally periodic since it is invariant under σ_1^2 and σ_2^2.

The following proposition states the fact that if a tile set admits some periodic tiling then it also admits a totally periodic tiling:

Proposition 29 *If a tile set admits an $\vec{r}$-periodic tiling for some $\vec{r} \neq \vec{0}$ then it also admits a totally periodic tiling.*

Proof. We can suppose that the tile set has radius-r neighborhood, for some $r \in \mathbb{Z}_+$. If not, we can add dummy neighbors as in the case of cellular automata until the neighborhood has this shape.

Let $t \in T^{\mathbb{Z}^2}$ be an $\vec{r}$-periodic tiling, for some $\vec{r} = (a, b)$. Let us assume without loss of generality that $b \neq 0$. The case $a \neq 0$ is symmetric. We can further suppose that $b \geq r$ because tiling t is $k\vec{r}$ -periodic for all integers k.

Consider rectangular regions whose widths and heights are $w = 2(|a| + r)$ and $h = b$, respectively. Let us partition the plane into such rectangular regions in a way that is invariant under translations by $\vec{r}$ and by $(w, 0)$, see Figure 21(a). Because also t is invariant under translations by $\vec{r}$ the sub-patterns of t in these rectangles are repeated as indicated in the figure. Since there are only a finite number of different sub-patterns with a fixed finite domain, it follows that some pattern A must appear twice on the same horizontal strip. It means that in tiling t there is an infinite strip S in the direction of $\vec{r}$ whose borders consist of copies of A, see Figure 21(b).

Because the width of the rectangle is sufficiently large the neighborhood of the right (left) half of each A in the strip S does not contain any cells to the left (right, respectively) of S. Consequently, an everywhere valid totally periodic tiling can be formed by repeating strip S horizontally as shown in Figure 21(c).

□

An interesting fact is that there exist tile sets that only allow non-periodic tilings. A tile set is called *aperiodic* if

(i) it admits some valid tilings, but

(ii) it does not admit any valid periodic tilings.

It was believed for a long time that aperiodic tile sets do not exist. This belief was refuted in 1966 by R.Berger who constructed a tile set that enforces non-periodicity. Note that valid tilings that are not periodic are easy to construct — the difficulty lies in the fact that all valid tilings must be non-periodic.

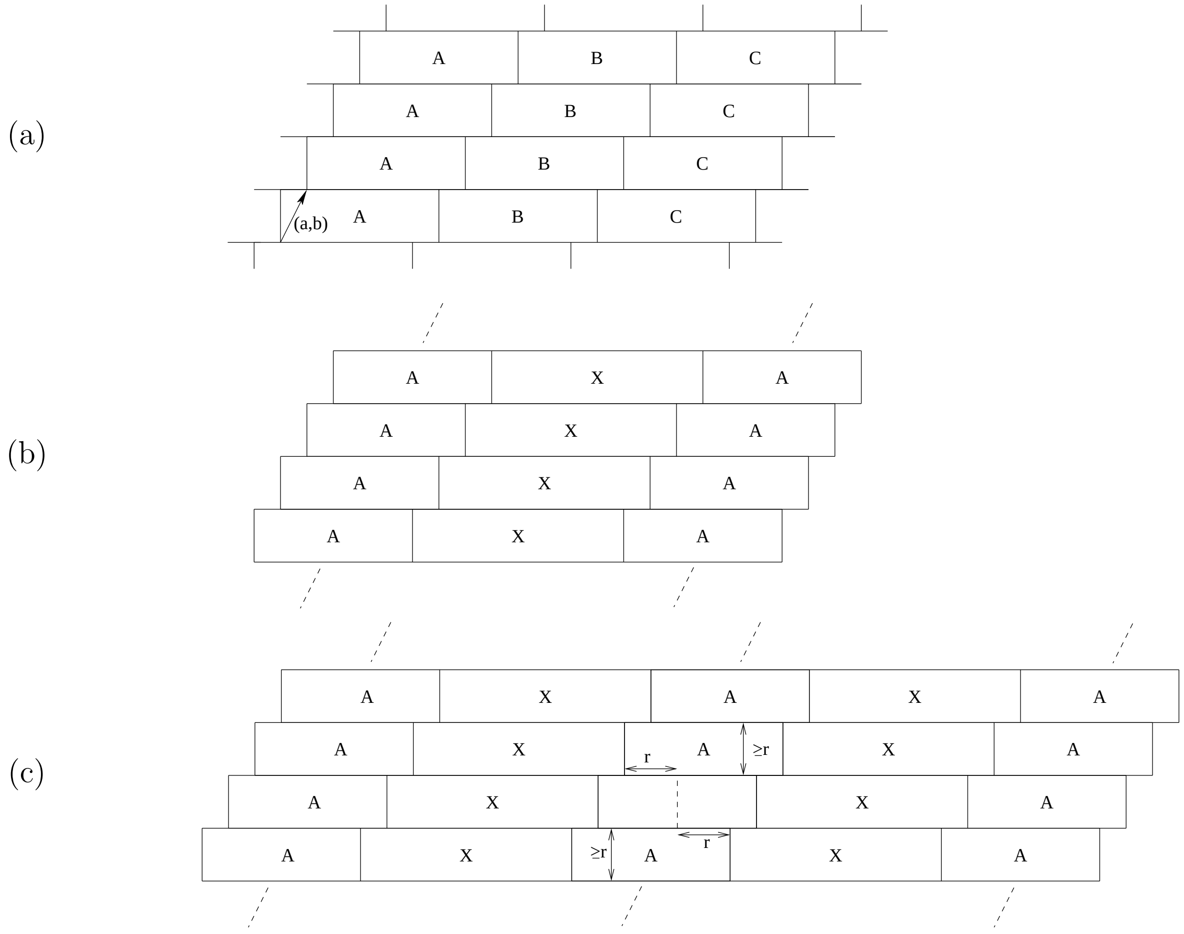


Figure 21: Illustrations for the proof of Proposition 29: (a) Partitioning of the plane into rectangles, (b) an infinite strip S in the direction of the period, and (c) a totally periodic tiling.

Proposition 30 *There exist aperiodic sets of Wang tiles.*

We skip the proof here. Aperiodic tile sets were constructed in the *Tiling and Patterns* course. For example, the 14 Wang tiles in Figure 22 were shown there to form an aperiodic set. In these tiles, rational numbers represent colors.

Aperiodic tile sets provide examples of cellular automata with unexpected properties:

Example 17. Let $\mathcal{T} = (T, N, R)$ be an aperiodic set of Wang tiles. Consider the following two-dimensional CA that uses the von Neumann neighborhood and whose state set is $T \cup \{q\}$ where $q \notin T$ is the quiescent state. Each cell checks if the states in its neighborhood are different from q and if they are, the cell checks whether the colors of its sides match with the adjacent tiles. If all four sides match in color then the cell does not change its state. In

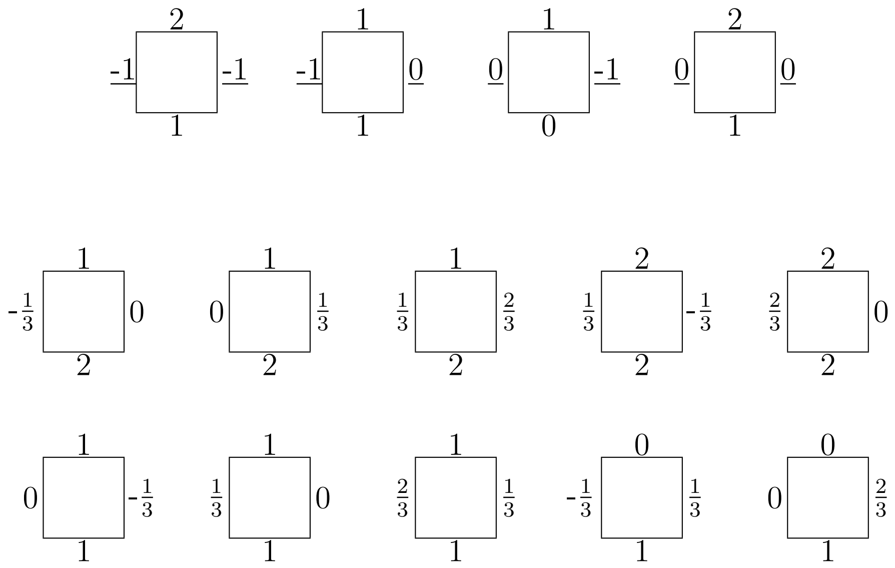


Figure 22: An aperiodic set of 14 Wang tiles.

all other cases the state becomes q.

In this CA, every totally periodic configuration becomes eventually quiescent. This follows from the fact that after the first application of the CA on a totally periodic configuration state q must appear since no valid periodic tiling exists. Once state q appears it starts spreading. Since the configuration is totally periodic it is clear that q's cover the whole plane after a finite number of applications of the CA function.

On the other hand, any valid (non-periodic) tiling provides a fixed point. These, together with the quiescent configuration, are the only fixed points, even the only temporally periodic configurations.

Let us then modify this CA by introducing a new intermediate state p that becomes q in one time step regardless of its neighbors. The local rule is modified so that a cell without a tiling error changes its state to p, and in all other cases the state becomes q. This CA has the property that the (totally periodic) configuration in which all states are equal to p has only non-periodic pre-images. Note that such a situation cannot happen in the one-dimensional case since any spatially periodic one-dimensional configuration that is not Garden-of-Eden has a spatially periodic pre-image.

Yet a third variant is a CA where state p always becomes q and vice versa, while a cell without a tiling error is unchanged. In all other cases the new state is q. This CA only has non-periodic fixed points. This is again a property that no one-dimensional CA can have. □

It may seem that tiles and tilings are only useful in constructing two- and higher dimensional CA. This is however not the case. Also one-dimensional cellular automata can be obtained from certain restricted types of tiles. We call a set T of Wang tiles *NW-deterministic* if for all $a, b \in T$, $a \neq b$, either the upper edges of a and b or the left edges of a and b have different colors. Then in every valid tiling each tile is uniquely determined by its left and upper neighbor. We define analogously *NE-, SW-* and *SE-deterministic* tile sets.

The aperiodic tile set in Figure 22 is not NW-deterministic since there are two tiles with labels 0 and 1 on their left and upper edges, respectively. However, there are aperiodic tile sets that are NW-deterministic. For example, Amman's aperiodic tile set from 1977 shown in Figure 23 is easily verified NW- deterministic (and simultaneously also SE-deterministic!) We skip the proof of its aperiodicity, see for example *Grünbaum, Shephard: Tilings and Patterns*. One can show that there even exist aperiodic tile sets that are deterministic in all four cornerwise directions (proof skipped):

Proposition 31 *There is an aperiodic set of Wang tiles that is NW-, NE-, SW- and SE-deterministic.* □

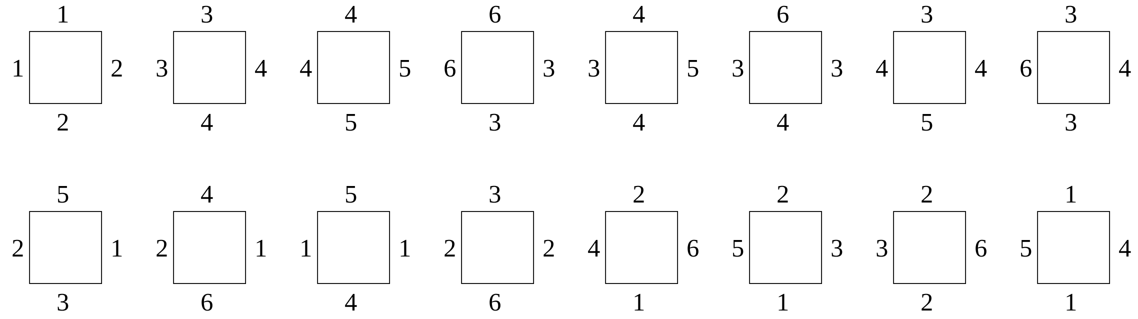


Figure 23: Amman's aperiodic NW- and SE-deterministic set of 16 Wang tiles.

Consider now a valid tiling of the plane by NW-deterministic tiles. Each tile is uniquely determined by its left and upper neighbor. Then tiles on each diagonal in the NE-SW direction locally determine the tiles on the next diagonal below it. If we interpret these diagonals as configurations of a CA then there is a local rule such that valid tilings are space-time diagrams of the CA, see Figure 24.

Example 18. Let $\mathcal{T} = (T, N, R)$ be the Amman's aperidic tile set in Figure 23. Consider the one-dimensional radius-$\frac{1}{2}$ CA whose state set is $S = T \cup \{q\}$ where $q \notin T$ is the quiescent state. The local rule $f : S^2 \longrightarrow S$ is

$$f(x, y) = \begin{cases} z, & \text{if } x, y, z \in T \text{ and they match as in Figure 24(a),} \\ q, & \text{otherwise.} \end{cases}$$

If c is a spatially periodic configuration then for some $t \in \mathbb{Z}_+$ configuration $G^t(c)$ contains state q, as otherwise a periodic tiling would exist. State q spreads to the left, so after a finite number of steps all cells are in state q. All spatially periodic configurations hence evolve into the quiescent configuration.

On the other hand, Amman's tile set admits a valid (non-periodic) tiling. Diagonals of the tiling provide configurations of a two-way infinite orbit in which no cell is in state q.

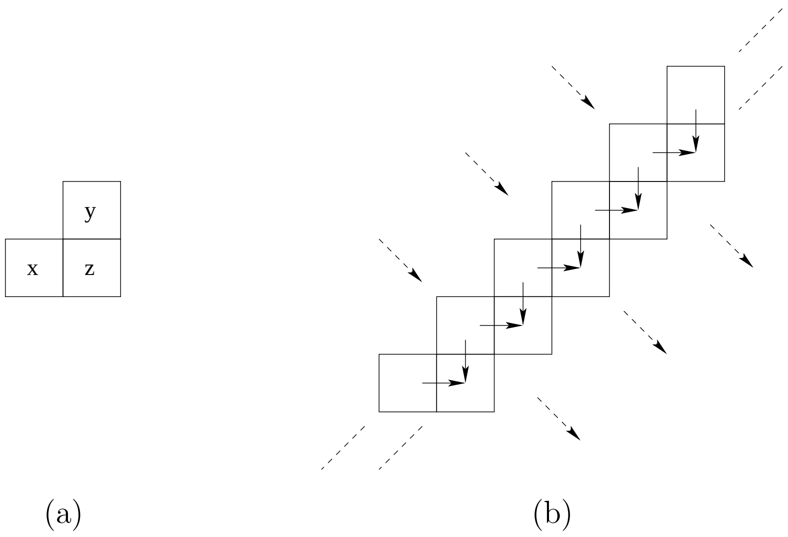


Figure 24: NW-deterministic sets of Wang tiles: (a) there is at most one matching tile z for any x and y, (b) diagonals of NW-deterministic tilings interpreted as configurations of one-dimensional CA.

Clearly such orbits cannot be periodic. The only temporally periodic configuration is the quiescent configuration, which is also a fixed point. □

A *directed tile* is a tile that is associated a *follower vector* $\vec{f} \in \mathbb{Z}^2$. Let $\mathcal{T} = (T, N, R)$ be a tile set, and let $F : T \longrightarrow \mathbb{Z}^2$ be a function that assigns tiles their follower vectors. We call $\mathcal{D} = (T, N, R, F)$ a set of directed tiles. Let $t \in T^{\mathbb{Z}^2}$. For every $\vec{p} \in \mathbb{Z}^2$ we call $\vec{p} + F(t(\vec{p}))$ the follower of $\vec{p}$ in t. In other words, the follower is the cell whose position relative to $\vec{p}$ is given by the follower vector of the tile in cell $\vec{p}$.

Sequence $\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_k$ where all $\vec{p}_i \in \mathbb{Z}^2$ is a (finite) *path* in t if

$$\vec{p}_{i+1} = \vec{p}_i + F(t(\vec{p}_i))$$

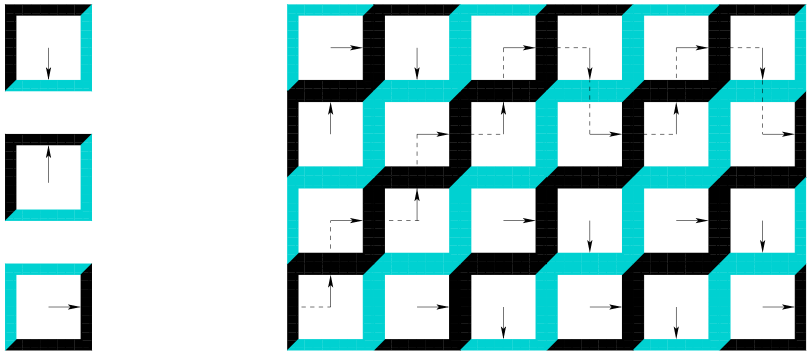
for all $1 \leq i < k$. In other words, a path is a sequence of cells such that the next cell is always the follower of the previous cell. One-way infinite and two-way infinite paths are defined analogously.

In our forthcoming considerations the follower of each tile is one of the four adjacent positions:

$$F(a) \in \{(\pm 1, 0), (0, \pm 1)\} \text{ for all } a \in T.$$

In this case the follower is indicated in drawings as a horizontal or vertical arrow over the tile, see Figure 25 for an example. From now on we assume only such followers.

A set of directed tiles is said to have the *plane-filling property* if it satisfies the following two conditions:



(a) (b)

Figure 25: (a) Three directed Wang tiles, and (b) a path on a valid tiling.

(a) It admits a valid tiling of the plane, and

(b) For any configuration $t \in T^{\mathbb{Z}^2}$, and for any one-way infinite path $\vec{p}_1, \vec{p}_2, \vec{p}_3, \ldots$ following the arrows on t, if the tiling in t is valid at $\vec{p}_i$ for all $i = 1, 2, 3, \ldots$, then there are arbitrarily large squares of cells such that all cells of the squares are on the path.

Intuitively the plane-filling property means that the simple device that moves over tiling t, verifies the correctness of the tiling it its present location, and moves on to the neighbor as indicated by the follower arrow in its present tile, necessarily eventually either finds a tiling error or covers arbitrarily large squares. Note that the plane-filling property does not assume that the tiling t is correct everywhere: as long as it is correct along a path the path must snake through larger and larger squares.

There exist tile sets that satisfy the plane filling-filling property. The proof of this result is also skipped:

Proposition 32 *There exists a set of directed Wang tiles that has the plane-filling property.* □

A proof for Proposition 32 constructs a set of Wang tiles such that the path that does not find any tiling errors is forced to follow the well known Hilbert-curve shown in Figure 26.

A periodic tile set can not satisfy the plane filling property: Every path on a valid tiling finds no tiling errors, but if the tiling is totally periodic then the path is also periodic, that is, the same directions are repeated periodically. But then the path is either a cycle or translation invariant, and hence it cannot contain all tiles of arbitrarily large squares.

Example 19. Let $\mathcal{D} = (T, N, R, F)$ be a directed tile set with the plane-filling property. Let us construct a two-dimensional CA whose state set is $S = T \times \{0, 1\}$, that uses the von-Neumann neighborhood and has the following local rule. For each $(a, b) \in S$ we call $a \in T$ the tile component and $b \in \{0, 1\}$ the bit component of the state. The tile components

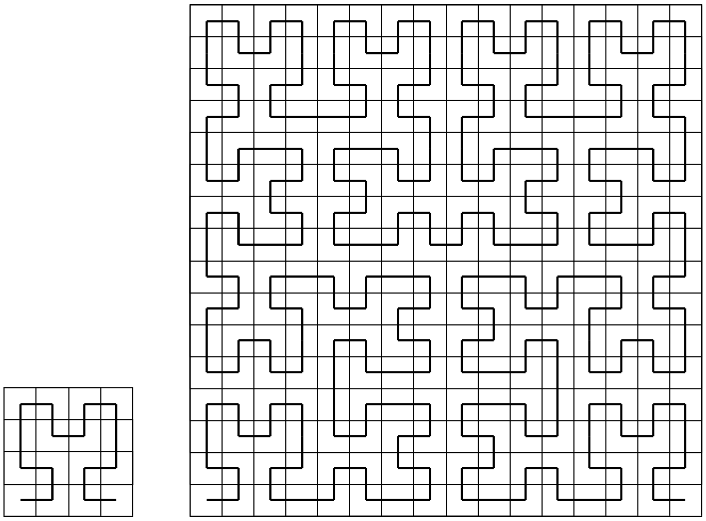


Figure 26: Fractions of the plane-filling Hilbert curve through 4 × 4 and 16 × 16 squares.

do not change. A cell checks if the tiling is valid at the cell. If the tiling is not valid then the bit component is not changed either. If the tiling is valid then the new bit component is $b_1 + b_2 \pmod 2$ where b_1 and b_2 are the bit components of the cell and its follower.

This CA is not reversible. If namely $c_0, c_1 \in S^{\mathbb{Z}^2}$ are such that the tile components in c_0 and c_1 form the same valid tiling, and the bit components of all cells in c_0 and c_1 are equal to 0 and 1, respectively, then both configurations c_0 and c_1 have the same image c_0.

But the CA is injective on periodic configurations: Suppose c_0 and c_1 are two different totally periodic configurations with the same image. Since the tile components are not changed we see that the tile components in c_0 and c_1 must be identical. Hence there is a cell $\vec{p}_1$ such that c_0 and c_1 have different bits at cell $\vec{p}_1$. Since these bits become identical in the next configuration, the tiling must be correct at $\vec{p}_1$ and c_0 and c_1 must have different bits in the follower position $\vec{p}_2$. We repeat the reasoning and obtain a one-way infinite sequence of positions $\vec{p}_1, \vec{p}_2, \vec{p}_3, \dots$ such that each $\vec{p}_{i+1}$ is the follower of $\vec{p}_i$, and the tiling is correct at each $\vec{p}_i$. Then, by the plane-filling property this path covers arbitrarily large squares. But this is in contradiction with the fact that c_0 and c_1 are periodic. □

The previous example proved the last remaining negative implication in Figure 9.

3 Algorithmic aspects

Next we turn to algorithmic aspects of cellular automata. These are two-fold: On one side, there are questions concerning cellular automata (e.g. is a given CA reversible? surjective?) that we want to find algorithms for. It turns out that many of these questions are undecidable, that is, no algorithm exists. On the other hand, cellular automata themselves are computation devices that can be used to solve algorithmic questions. We investigate computational universality in cellular automata.

First we review some basic concepts of the theory of computability.

3.1 Algorithms, semi-algorithms and undecidability

Intuitively, an algorithm is a mechanical procedure, specified by a finite set of instructions, to solve some well-defined computational problem. The problem has an input *instance*, and the algorithm, after a finite number of processing steps, must return the answer to the problem for the given instance. We focus on algorithms for *decision problems*. These are problems where the answer for each instance is either "yes" or "no". For example, the question whether a given cellular automaton is reversible is a decision problem. An instance of this problem is a cellular automaton, and the answer is "yes" or "no" depending on whether the CA is reversible or not. We will call these positive and negative instances of the problem, respectively.

Computer programming is common place. There is widespread acceptance that algorithms are processes that can be implemented as programs in some standard programming language, say for example, in C. So in the following discussion algorithms for decision problems will be understood as computer programs that take the instance of the problem as input, and return answer "yes" or "no". Rather than providing actual syntactically correct programs, algorithms will be described in plain English in sufficient details so that an experienced programmer easily can implement the algorithm on a computer.

A *semi-algorithm* for a decision problem is an algorithm like process that correctly returns value "yes" for positive instances, but on negative instances the semi-algorithm does not return any value, i.e. it runs for ever without ever halting. Since "yes" is the only possible output from a semi-algorithm we can simply state that the input instance is positive if and only if the semi-algorithm halts. Notice that there is non-symmetry between the positive and negative instances. So a semi-algorithm for reversibility of cellular automata is not a semi-algorithm for non-reversibility. The decision problem obtained by swapping the negative and positive instances is called the *complement* of the original problem.

If an algorithm exists for a decision problem then we say that the decision problem is *decidable*. Otherwise it is *undecidable*. If a semi-algorithm exists then the problem is called *semi-decidable*. Note the following obvious facts:

(i) If a problem is decidable then also the complement problem is decidable.

(ii) Every decidable problem is also semi-decidable.

(iii) If a problem and its complement are both semi-decidable then the problem is decidable.

For observation (iii) note that the two semi-algorithms for the problem and its complement can be executed concurrently until one of them returns the answer.

Example 20. The decision problem

1D Reversibility
Instance: One-dimensional cellular automaton A
Problem: Is A reversible ?

is decidable. An algorithm based on the pair graph was described in Section 2.6. Also the problem

1D Surjectivity
Instance: One-dimensional cellular automaton A
Problem: Is A surjective ?

was shown decidable by the pair graphs technique. The higher dimensional variants are more complicated.

2D Reversibility
Instance: Two-dimensional cellular automaton A
Problem: Is A reversible ?

is semi-decidable: A semi-algorithm enumerates one-by-one all two-dimensional cellular automata X that have the same state set as A, and tests for each candidate whether $X \circ A$ computes the identity function. If such an X is found then the semi-algorithm halts, indicating that the input A is reversible, otherwise the process continues without ever halting. Observe that one can effectively construct candidates X one-by-one, form the composition $X \circ A$ and test for equivalence with the identity function.

Later we see that 2D Reversibility is undecidable, so no semi-algorithm exists for recognizing non-reversible CA. The decision problem

2D Surjectivity
Instance: Two-dimensional cellular automaton A
Problem: Is A surjective ?

is not semi-decidable, but its complement is. A semi-algorithm for negative instances is based on looking for an orphan: it enumerates all finite patterns p and checks for each p whether it is an orphan or not. If an orphan is found then the process halts, indicating that the instance is not a surjective CA. On surjective CA the search for an orphan continues without ever halting. □

In practice, the input to a program is an encoding of an instance as a string. This is clear in the programming context, since the input to the program will be keyed in from a keyboard, or read in from a file. In any case, inside the computer, the input will be represented as a sequence of bits, that is, as a word over the two letter alphabet $\{0, 1\}$. Each decision problem can now be associated a language over the encoding alphabet: the language contains all those words that are encodings of positive instances. Then solving the decision problem is equivalent to determining if a given word belongs to the corresponding language. Notice that there are actually three types of words: encodings of positive instances, encodings of negative instances and those words that are not valid encodings of any instances. In the following we always assume that the encoding is effective in the sense that there is an algorithm to determine if a given word is a valid encoding. Also the way the instances are encoded should be natural so that the decision problem is not solved or otherwise affected by the encoding. (For example adding letter "+" or "-" in front of the word to indicate whether it is an encoding of a positive or negative instance is not acceptable, since then the decision problem must be solved already during the encoding process!) Actually all results concerning the decidability of a decision problem are relative to the encoding used, but all reasonable encodings are equivalent with each other in the sense that there are algorithms to convert between the encodings, which means that the decidability/semi-decidability status is unaffected by the choice of the encoding.

The decision problem whether a given word belongs to language $L \subseteq \Sigma^*$ is called the *membership problem* of language L. Language L is called *recursive* if its membership problem is decidable, and it is called *recursively enumerable* (or r.e. for short) if its membership problem is semi-decidable. Clearly a decision problem is decidable (or semi-decidable) if and only if the membership problem for the corresponding language is recursive (or recursively enumerable, respectively).

The number of different algorithms and semi-algorithms is countable. Each algorithm can namely be represented as a finite string (e.g. its source code in programming language C), and the number of different strings over any alphabet is countable. This means that there are only countably many recursive and recursively enumerable languages over any alphabet Σ. On the other hand, there are uncountably many different languages $L \subseteq \Sigma^*$, which implies that many languages are not recursive (or even recursively enumerable). Their membership problem is not decidable (or semi-decidable), so we see that there are many undecidable decision problems. This is not surprising, but what is more interesting is that we can prove some individual problems undecidable, and that many of these undecidable problems are quite natural and appear in many contexts.

To obtain our first undecidable decision problem we turn to (semi-)algorithms whose input is a semi-algorithm A. This of course means that the input is an encoding of A as, say, a binary string representing the source code for A in language C. To clarify the distinction we'll denote by $\langle A\rangle$ the encoding of A over the binary alphabet. An algorithm call to A with input w will be denoted by $A(w)$.

The decision problem that we first prove undecidable is the following:

Semi-algorithm halting
Instance: A semi-algorithm $\langle A\rangle$ and an input string $w \in \Sigma^*$
Problem: Does A halt on input w ?

Proposition 33 *The decision problem* Semi-algorithm halting *is undecidable.*

Proof. Suppose the contrary: There is an algorithm `Halt` that solves problem Semi-algorithm halting. In other words, for any binary strings $\langle A\rangle$ and w algorithm call `Halt`$(\langle A\rangle, w)$ returns value "yes" if A is a semi-algorithm that halts on input w, and `Halt`$(\langle A\rangle, w)$ returns value "no" otherwise. Using `Halt` we can construct the following semi-algorithm:

```
Diag(⟨A⟩):
   if Halt(⟨A⟩,⟨A⟩) returns ''no'' then halt otherwise loop forever
```

The semi-algorithm Diag takes as input a binary string $\langle A\rangle$. Using an algorithm call to Halt it determines whether the input is the encoding of a semi-algorithm A that halts with input $\langle A\rangle$, i.e. on its own encoding. If Halt($\langle A\rangle, \langle A\rangle$) returns value ”no” then Diag($\langle A\rangle$) halts and if Halt($\langle A\rangle, \langle A\rangle$) returns ”yes” then Diag($\langle A\rangle$) enters an infinite loop and never halts.

Consider the algorithm call Halt(⟨Diag⟩, ⟨Diag⟩). If it returns ”yes” then Diag(⟨Diag⟩) should halt. But following the definition of Diag this means that Halt(⟨Diag⟩, ⟨Diag⟩) returned ”no” when Diag made that call, a contradiction.

On the other hand, if Halt(⟨Diag⟩, ⟨Diag⟩) returns ”no” then Diag(⟨Diag⟩) should not halt. That happens if and only if Halt(⟨Diag⟩, ⟨Diag⟩) returns ”yes”, again a contradiction. Both cases lead to a contradiction, so algorithm Halt does not exist. □

Note that Semi-algorithm halting is semi-decidable, so its complement problem is not. To see the semi-decidability consider the semi-algorithm that on inputs $\langle A\rangle$ and w starts to simulate A on input w step-by-step. Halting of this process correctly identifies the positive instances of Semi-algorithm halting.

The proof of Proposition 33 (due to A.Turing) can be understood as a variant of the diagonal argument that Cantor used to prove that certain sets are uncountable. Imagine an infinite two-dimensional matrix M whose rows and columns are indexed by $\langle A\rangle$ for all semi-algorithms A, and where

$$M[\langle A\rangle, \langle B\rangle] = \begin{cases} 0, & \text{if } A \text{ does not halt on input } \langle B\rangle, \\ 1, & \text{if } A \text{ halts on input } \langle B\rangle. \end{cases}$$

Consider the diagonal elements $M[\langle A\rangle, \langle A\rangle]$ of the matrix. Reading the diagonal and swapping each value $0 \leftrightarrow 1$ creates a sequence that is different from every row of the matrix. This means that there is no semi-algorithm enlisted that would halt on exactly those inputs $\langle A\rangle$ for which A does not halt on input $\langle A\rangle$. But such a semi-algorithm should exist if Semi-algorithm halting is decidable.

Once we established the undecidability of Semi-algorithm halting we can use *reductions* to obtain new undecidable problems.

Example 21. Let us prove that is undecidable whether a given semi-algorithm halts on some input.

Sometimes halting
Instance: A semi-algorithm $\langle A\rangle$
Problem: Does A halt on some input w ?

Suppose there is an algorithm S that solves Sometimes halting. Let us describe an algorithm Halt that then solves Semi-algorithm halting: Let $\langle A\rangle$ and w be arbitrary inputs to Halt. First string $\langle B\rangle$ is created where B is a semi-algorithm that takes one input and works as follows: B checks whether its input is equal to w. If it is not then it enters an infinite loop. If the input is w then it starts A with input w. Clearly B halts with input w if A halts with input w and it does not halt on any input if A does not halt on input w.

Note that this B can be effectively created for any given A and w, that is, there is an algorithm that outputs $\langle B \rangle$ when it gets $\langle A \rangle$ and w as input.

Algorithm `Halt` next takes $\langle B \rangle$ that it created and gives it as input to the hypothetical algorithm `S`. From the answer it can conclude whether A halts on w.

We have described algorithm `Halt` that solves problem Semi-algorithm halting. This is in contraction to Proposition 33, which means that the hypothetical algorithm `S` does not exist. □

The previous example illustrates the idea of *Turing reductions*: If we want to prove that decision problem P is undecidable we describe an algorithm for some known undecidable problem U that makes (possibly several) calls to a hypothetical algorithm for P. Since U is previously known to be undecidable such algorithm cannot exist. Therefore the hypothetical algorithm for P cannot exist either.

Example 21 shows actually a reduction of more restricted type: A *many-one reduction* is an algorithm that modifies an arbitrary input u for a known undecidable problem U into an equivalent instance p of decision problem P. By equivalent it is meant that p is a positive instance of P if and only if u is a positive instance of U. The existence of such reductions shows that P is undecidable, because otherwise the reduction and an algorithm for P would solve U. Notice that many-one reductions are special types of Turing reductions where the call to the hypothetical algorithm for P is done only once, as the last step.

Finally, reductions can be used also to show that some decision problem P is not even semi-decidable. In this case one starts with a hypothetical semi-algorithm `S` for for P, and shows that there is then a semi-algorithm for some problem U that is previously known not to be semi-decidable. In case of Turing reductions this semi-algorithm is allowed to make any number of calls to `S`, while in many-one reductions only one call is made at the end.

3.2 Turing machines

Turing machines are stripped down toy computers that are extremely simple in their definition and step-by-step operations, but nevertheless powerful enough to simulate arbitrary algorithms. It is actually common to take them as the formal definition of an algorithm. (In this course a different approach was taken and we rather defined algorithms in terms of computer programs, since for most people it is easier to accept these as the proper model of computation.)

There are two reasons for us to introduce Turing machines at this stage:

(i) We use reductions to establish new undecidability results. Direct reductions from decision problems involving semi-algorithms into simple questions concerning tilings and cellular automata are very laborious as they involve converting given semi-algorithms into corresponding tile sets or CA. Reductions from simple devices such as Turing machines are shorter and easier. So a good approach is to first prove one question concerning Turing machines (the halting problem of Turing machines) undecidable, and to make further reduction from this simple set-up.

(ii) We want to show that cellular automata can perform arbitrary computations. This requires simulating arbitrary (semi-)algorithms on cellular automata. As in (i), it is much easier to show how to simulate a simple device such as a Turing machine rather than an algorithm given in terms of, say, a C-program.

A *deterministic Turing machine* is specified by the following items:

- A finite *state set* Q,
- *initial*, *accepting* and *rejecting* states $q_0, q_a, q_r \in Q$, respectively, where q_a and q_r are both called *halting* states and $q_a \neq q_r$,
- a finite *tape alphabet* Γ,
- an *input alphabet* $\Sigma \subset \Gamma$,
- a *blank* tape symbol $b \in \Gamma \setminus \Sigma$, and
- a *transition function* $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{-1, 1\}$. For all $\gamma \in \Gamma$ we must have $\delta(q_a, \gamma) = (q_a, \gamma, 1)$ and $\delta(q_r, \gamma) = (q_r, \gamma, 1)$.

The machine consists of a *tape* and a *control unit.* The tape is a two-way infinite sequence of cells, each capable of storing a letter from the tape alphabet Γ. The tape positions are indexed by $\mathbb{Z}$. The content of the tape at any given time is described by a function $t \in \Gamma^{\mathbb{Z}}$ where $t(i) \in \Gamma$ is the symbol in cell i, for any $i \in \mathbb{Z}$.

The control unit is a finite state automaton that moves on the tape and is able to read the symbol at its present location on the tape. Triplets

$$(q, i, t) \in Q \times \mathbb{Z} \times \Gamma^{\mathbb{Z}}$$

are *instantaneous descriptions* or the *configurations* of the Turing machine. They contain all information about the present state of the machine: its state q, location i on the tape and the content t of the tape. Configurations (q, i, t) where $q = q_a$ or $q = q_r$ are called accepting and rejecting, respectively. In both cases we say that the Turing machine halts.

In one time step the Turing machine – depending on the state of the control unit and the tape symbol in the cell it reads – changes the state, replaces the tape symbol by a new one and moves one position to the left or right on the tape, as specified by the transition function δ. More precisely, configuration (q, i, t) becomes $(q', i + d, t')$ if $\delta(q, t(i)) = (q', \gamma, d)$ and $t'(i) = \gamma$ and $t'(j) = t(j)$ for all $j \neq i$. We denote this move by

$$(q, i, t) \vdash (q', i + d, t').$$

The reflexive, transitive closure of $\vdash$ is denoted by $\vdash^*$, that is,

$$(q, i, t) \vdash^* (q', i', t')$$

if and only if (q', i', t') can be reached from (q, i, t) by executing zero or more Turing machine steps.

Turing machines are used to recognize languages over input alphabet Σ. For any $w \in \Sigma^*$ we denote by $t_w \in \Gamma^{\mathbb{Z}}$ the tape content where we have written word w in cells $1, 2, \ldots, |w|$ and all other cells contain the blank symbol b. The machine *accepts* word w if and only if

$$(q_0, 1, t_w) \vdash^* (q_a, i, t)$$

for some $i \in \mathbb{Z}$ and $t \in \Gamma^{\mathbb{Z}}$, that is, the machine halts in the accepting state q_a. Otherwise the word is *rejected*. Note that rejection can happen in two different ways: Either the machine halts in the rejecting state q_r or it never halts. The set of accepted words is the language $L(M) \subseteq \Sigma^*$ recognized by Turing machine M.

Example 22. Consider the following Turing machine:

- $Q = \{A, B, q_a, q_r\}$ where $A = q_0$ is the initial state and q_a and q_r are the accepting and rejecting halting states.
- $\Gamma = \{0, 1\}$ where 0 is the blank symbol and $\Sigma = \{1\}$.
- The transition function δ is as follows:

$$\begin{array}{rcl} (A, 0) & \mapsto & (B, 1, 1) \\ (A, 1) & \mapsto & (B, 1, -1) \\ (B, 0) & \mapsto & (A, 1, -1) \\ (B, 1) & \mapsto & (q_a, 1, 1) \end{array}$$

(and q_a and q_r are halting.)

With the empty input word (=on the initially blank tape) the machine halts after six moves:

$$\begin{array}{rl} & \ldots 0\;0\;0\;0\;\overset{A}{0}\;0\;0\;0\;.. \\ \vdash & \ldots 0\;0\;0\;0\;1\;\overset{B}{0}\;0\;0\;.. \\ \vdash & \ldots 0\;0\;0\;0\;\overset{A}{1}\;1\;0\;0\;.. \\ \vdash & \ldots 0\;0\;0\;\overset{B}{0}\;1\;1\;0\;0\;.. \\ \vdash & \ldots 0\;0\;\overset{A}{0}\;1\;1\;1\;0\;0\;.. \\ \vdash & \ldots 0\;0\;1\;\overset{B}{1}\;1\;1\;0\;0\;.. \\ \vdash & \ldots 0\;0\;1\;1\;\overset{q_a}{1}\;1\;0\;0\;.. \end{array}$$

This example is a two-state *busy beaver* over the two letter alphabet: It runs on the initially blank tape for the longest possible time among all Turing machines that have two non-halting states and two tape letters, and that halt on the initially blank tape.

The numbers of moves by the three and four state busy beavers over the two letter alphabet are 21 and 107, respectively. The number of moves by five and six state busy beavers are not known but they are at least 47176870 and 10^{2879}, respectively. □

It is easy to see that the language recognized by a Turing machine is recursively enumerable: For any Turing machine one can construct a semi-algorithm that simulates the Turing machine step-by-step on any given input word until the word gets accepted. If the word is not accepted then the semi-algorithm does not halt.

What is more complicated to see is that every recursively enumerable language is recognized by a Turing machine. This is based on the fact that Turing machines — despite the simplicity of their individual moves — can simulate the computation steps of arbitrary (semi-)algorithms. The proof of this fact is beyond the scope of the present notes:

Proposition 34 *There is an algorithm that for any given (semi-)algorithm A whose input is a word over alphabet Σ constructs a Turing machine M with the same input alphabet Σ such that on every input $w \in \Sigma^*$*

- *if A returns "yes" then M eventually enters its accepting state f,*
- *if A returns "no" then M eventually enters its rejecting state r, and*
- *if A does not halt then M does not halt either.*

□

From the proposition we directly get the following corollary:

Corollary 35 *A language is recursively enumerable if and only if there is a Turing machine that recognizes it.* □

Consider then Turing machines that have the property that they halt on every input word. Such Turing machines only recognize recursive languages since an algorithm can simulate them. Also the converse is true, as implied by Proposition 34:

Corollary 36 *A language is recursive if and only if it is recognized by a Turing machine that halts on every input word.* □

Note: It is customary in the field of theoretical computer science to take Turing machines as the formal definition of (semi)algorithms. In this case the above corollaries are definitions, and theorems can be stated to the effect that (semi)algorithms defined under other models (C-programs, for example) are equivalent to Turing machines. We did the process in reverse:

(semi)algorithms were defined in terms of C-programs, and the Corollaries above tie them to Turing machines.

Using Proposition 34 one can reduce decision problems concerning semi-algorithms into analogous questions concerning Turing machines. For example, undecidability of Semi-algorithm halting (Proposition 33) implies that it is undecidable if a given Turing machine halts on a given input word. Undecidability of Sometimes halting (Example 21) implies that it is undecidable if a given Turing machine halts on some input. The most elementary problem concerning Turing machines asks whether a given Turing machine halts when it is started on the blank tape:

TM halting on blank tape
Instance: A Turing machine M
Problem: Does M accept the empty input word ?

Proposition 37 *The decision problem* TM halting on blank tape *is undecidable. More precisely, it is semi-decidable while the complement problem is not semi-decidable.*

Proof. The semi-decidability follows from an effective step-by-step simulation of the Turing machine until (if ever) it halts. To prove undecidability we reduce Semi-algorithm halting. For any given semi-algorithm A and its input w we can effectively construct a semi-algorithm B that totally ignores its input u and simply calls A with input w. Clearly this B halts on all inputs (including the empty word) if A halts on input w and does not halt on any input (including the empty word) if A does not halt on input w. Using Proposition 34 we effectively construct a Turing machine M that is equivalent to B. Machine M hence halts on the initially empty tape if and only if A halts on w. We have many-one reduced the undecidable problem Semi-algorithm halting into TM halting on blank tape. □

Example 23. Consider the following busy beaver function $BB : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$ where $BB(n,m)$ is the largest number t such that there is a Turing machine with n non-halting states and m tape symbols that makes t moves on the blank tape and then halts. In Example 22 we discussed this function and mentioned that $BB(2,2) = 6$, $BB(3,2) = 21$, $BB(4,2) = 107$, $BB(5,2) \geq 47176870$ and $BB(6,2) \geq 10^{2879}$. It follows from Proposition 37 that BB grows faster than any function that can be computed. Namely, if there would be an algorithm A that outputs for any given $n \in \mathbb{N}$ and $m \in \mathbb{N}$ a value t such that $BB(n,m) \leq t$ then this algorithm could be used to effectively solve TM halting on blank tape: For any given Turing machine M we would namely obtain using algorithm A a number t such that if M halts on the blank tape, it does so within t steps. Then it is an easy matter to make the first t moves of M to see if it halts. It follows hence from Proposition 37 that algorithm A cannot exist.

Such functions that grow faster than any computable function are associated with all undecidable problems we encounter. □

3.3 Undecidability in tiles

In order to proof undecidability results for cellular automata it is convenient to take an intermediate step and prove certain questions concerning Wang tilings undecidable. The proofs are based on the fact that valid tilings can be forced to contain a complete simulation of a given Turing machine. To any given Turing machine M we associate the Wang tiles shown in Figure 27, and we call these tiles the *machine tiles* of M. Note that in the illustrations, instead of colors, we use labeled arrows on the sides of the tiles. Two adjacent tiles match if and only if an arrow head meets an arrow tail with the same label. Such arrow representation can be converted into the usual coloring representation of Wang tiles by assigning to each arrow direction and label a unique color.

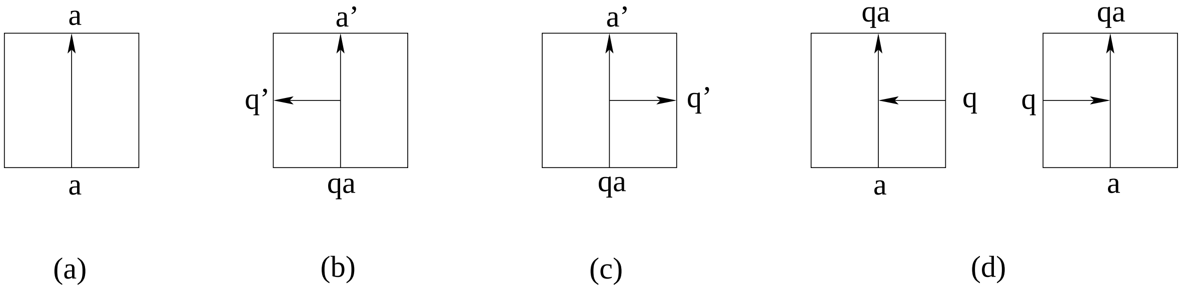


Figure 27: Machine tiles associated to a Turing machine.

The machine tiles of M contain the following tiles:

(i) For every tape letter $a \in \Gamma$ a *tape tile* of Figure 27(a),

(ii) For every tape letter $a \in \Gamma$ and non-halting state $q \in Q \setminus \{q_a, q_r\}$ an *action tile* of Figure 27(b) or (c). Tile (b) is used if

$$\delta(q, a) = (q', a', -1)$$

and tile (c) is used if

$$\delta(q, a) = (q', a', +1).$$

(iii) For every tape letter $a \in \Gamma$ and non-halting state $q \in Q \setminus \{q_a, q_r\}$ two merging tiles shown in Figure 27(d).

The idea of the tiles is that a configuration of the Turing machine M is represented as a row of tiles in such a way that the cell currently scanned by M is represented by an action tile, its neighbor where the machine moves into has a merging tile and all other tiles on the row are tape tiles. If this is a row of a valid tiling then it is clear that the rows above must be similar representations of subsequent configurations in the Turing machine computation, until the machine halts.

Let us begin with the seeded variant of the tiling problem. In this decision problem we are given a finite set T of Wang tiles and one *seed tile* $s \in T$, and we ask whether there exists a valid tiling of the plane where s appears at least once:

Tiling problem with the seed tile
Instance: A finite set T of Wang tiles and a seed tile $s \in T$
Problem: Does there exist a valid tiling t of the plane such that $t(0,0) = s$?

Proposition 38 *The* Tiling problem with the seed tile *is undecidable. More precisely, it is not semi-decidable while the complement problem is semi-decidable.*

Proof. The semi-decidability of the complement problem follows from the following semi-algorithm: For $r = 1, 2, 3, \ldots$ try all tilings of the radius r square around the origin to see if there is a valid tiling of the square such that the origin contains the seed tile s. If for some r such a tiling is not found then halt and report that there is no tiling containing the seed tile.

Consider then the undecidability. We reduce the TM halting on blank tape, proved undecidable in Proposition 37. For any given Turing machine M we construct the machine tiles of Figure 27 as well as the four tiles shown in Figure 28. These are the blank tile and three initialization tiles. They initialize all tape symbols to be equal to blank b, and the Turing machine to be in the initial state q_0. The second initialization tile is chosen as the seed tile s.

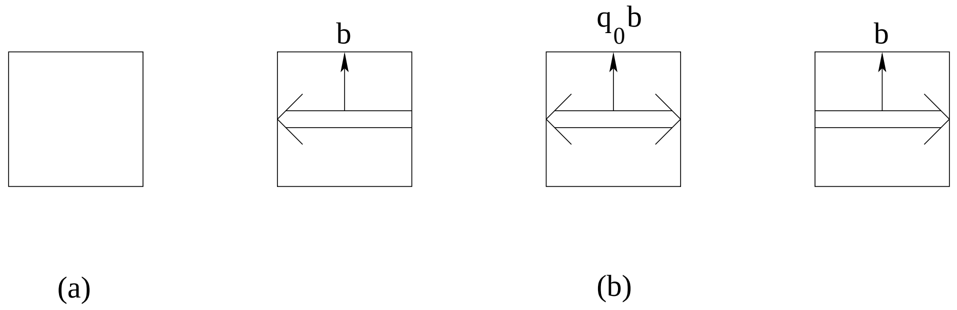


Figure 28: (a) the blank tile, and (b) three initialization tiles.

Let us prove that a valid tiling containing a copy of the seed tile exists if and only if the Turing machine M does not halt when started on the blank tape:

"$\Longleftarrow$": Suppose that the Turing machine does not halt on the blank tape. Then a valid tiling exists where one horizontal row is formed with the initialization tiles, all tiles below this row are blank, and the rows above the initialization row contain consecutive configurations of the Turing machine.

"$\Longrightarrow$": Suppose that a valid tiling containing the middle initialization tile exists. The seed tile forces its row to be formed from the initialization tiles, representing the initial configuration of the Turing machine on the blank tape. The machine tiles force following horizontal rows above the seed row to contain the consecutive configurations of the Turing machine. There is no tile containing a halting state so the Turing machine does not halt — otherwise a valid tiling could not be formed. □

In the following variant we are given a Wang tile set T and specify one tile $B \in T$ as the *blank tile.* The blank tile has all four sides colored by the same color. A *finite tiling* is a tiling where only a finite number of tiles are non-blank. A finite tiling where all tiles are blank is called *trivial.*

Finite tiling problem
Instance: A finite set T of Wang tiles and a blank tile $B \in T$
Problem: Does there exist a valid finite tiling that is not trivial ?

Proposition 39 *The* Finite tiling problem *is undecidable. It is semi-decidable while its complement is not semi-decidable.*

Proof. For semi-decidability notice that we can try all valid tilings of larger and larger squares until we find a tiling of a square where all tiles on the boundary are blank, while some interior tile is different from the blank tile. If such a tiling is found then the semi-algorithm halts, indicating that a valid, finite, non-trivial tiling exists.

To prove the undecidability we reduce the problem TM halting on blank tape. For any given Turing machine M we construct the machine tiles of Figure 27 as well as the blank tile, boundary tiles and the halting tiles shown in Figure 29.

The halting tiles of Figure 29(b) are constructed for all tape letters $a \in \Gamma$ and both halting states $q = q_a$ and $q = q_r$. The purpose of the halting tiles is to erase the Turing machine from the configuration once it halts. The lower border tiles of Figure 29(c) initialize the configuration to consist of the blank tape symbol b and the initial state q_0. The top border tiles are made for every tape symbol $a \in \Gamma$. They allow the absorption of the configuration as long as the Turing machine has been erased. The border tiles on the sides are labeled with symbols L and R to identify the left and the right border of the computation area.

Let us prove that the tile set admits a valid, finite, non-trivial tiling if and only if the Turing machine halts on the empty tape.

"$\Longleftarrow$": Suppose that the Turing machine halts on the blank tape. Then a tiling exists where the boundary tiles isolate a finite portion of the plane (a "board") for the simulation of the Turing machine, the bottom tiles of the board initialize the Turing machine on the blank tape, and inside the board the Turing machine is simulated until it halts. After halting only tape tiles are used until they are absorbed by the topmost row of the board. If the board is made sufficiently large the entire computation fits inside the board, so the tiling is valid. All tiles outside the board are blank so the tiling is finite.

"$\Longrightarrow$": Suppose then that a finite, non-trivial tiling exists. The only non-blank tiles with blank bottom edge are the lower border tiles of Figure 29(c), so the tiling must contain a

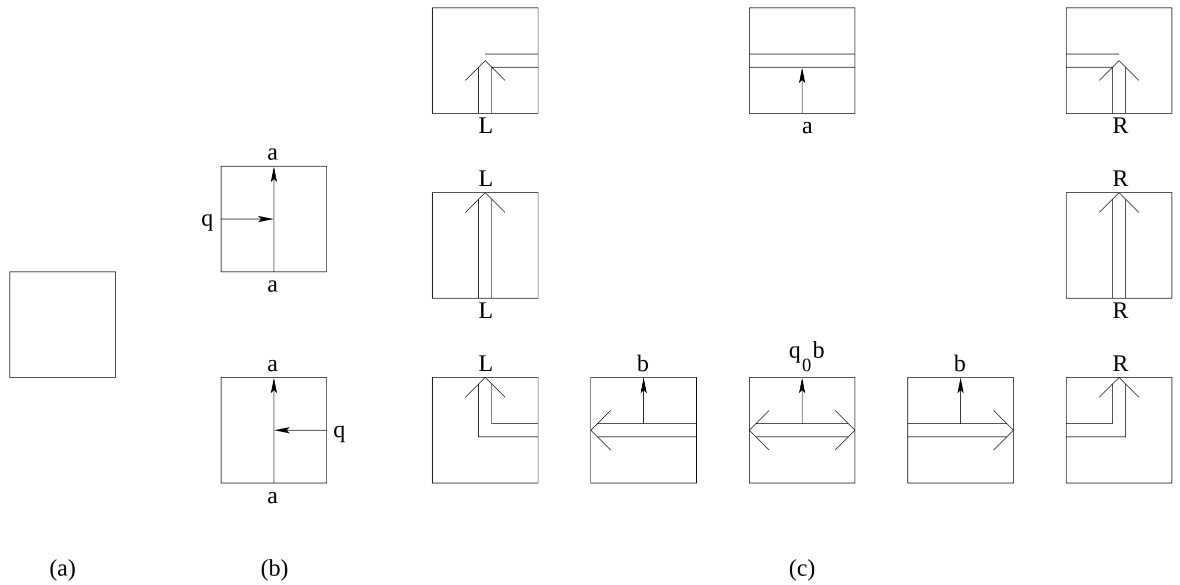


Figure 29: (a) the blank tile, (b) halting tiles, and (c) border tiles.

lower border tile. Horizontal neighbors of lower border tiles are lower border tiles, so we see that the only way to have a finite tiling is to have a contiguous lower border that ends at both sides in a corner tile where the border turns upwards. The vertical borders must again — due to the finiteness of the tiling — end at corners where the top border starts. All in all we see that the boundary tiles are forced to form a rectangular board.

The lower boundary of the board initializes the Turing machine configuration on the blank tape, and the rows above it are forced by the machine tiles to simulate consecutive configurations of the Turing machine. Because the Turing machine state symbol is not allowed to touch the side or the upper boundary of the board, the Turing machine must be erased by a halting tile, i.e. the Turing machine must halt. □

The two tiling problem variants above have been fairly simple to prove undecidable because we have been given the tools to force the initialization of the Turing machine in valid tilings. The next problem is is more complicated:

Tiling problem
Instance: A finite set T of Wang tiles
Problem: Does there exist a valid tiling ?

The Tiling problem was proved undecidable by R.Berger in 1966, in the same piece of work where the first aperiodic tile set was constructed. This is not coincidence: if aperiodic tile sets would not exist then the Tiling problem would be decidable. One could namely try all tilings of larger and larger squares until one of the following two conditions is satisfied:

- A square is found that cannot be tiled, or

- A tiling of a square is found where the top and the bottom colors match and left and the right sides match.

The first condition happens if and only if no valid tiling of the plane exists, and the second condition is satisfied if and only if a periodic tiling exists. If aperiodic tile sets did not exist then one of the two conditions would always be satisfied, and this would provide then an algorithm for the Tiling problem. But aperiodic tile sets fail to satisfy either of the two conditions.

We state Berger's result without a proof (it has been proven in the *Tilings and Patterns* class). In fact, an even stronger variant will be used in the next section:

NW-deterministic tiling problem
Instance: A NW-deterministic set T of Wang tiles
Problem: Does there exist a valid tiling ?

Proposition 40 *The* NW-deterministic tiling problem *is undecidable. More precisely, the problem is not semi-decidable but its complement is semi-decidable.* □

Finally, as the last variant of the tiling problem we state without a proof the undecidability of the problem that asks whether a given tile set admits a periodic tiling:

Periodic tiling problem
Instance: A finite set T of Wang tiles
Problem: Does there exist a valid periodic tiling ?

Proposition 41 *The* Periodic tiling problem *is undecidable. More precisely, the problem is semi-decidable while its complement is not.* □

3.4 Undecidable questions concerning cellular automata

Next we turn to our main topic and prove several questions concerning cellular automata undecidable. The input instance to these decision problems include a cellular automaton. The problems typically concern properties of its global function G. We usually state that the instance of a problem is a CA function G, but one should keep in mind that, in fact, the input instance is given in terms of the local rule that specifies G.

We start with a problem of deciding if a given CA function has trivial behavior. A CA function G is called *nilpotent* if all configurations eventually lead to the quiescent configuration. In fact, then there is a constant $n \in \mathbb{Z}_+$ such that $G^n(c)$ is the quiescent configuration for all $c \in S^{\mathbb{Z}^d}$. To see this, consider a *transitive* configuration $t \in S^{\mathbb{Z}^d}$ that contains a copy of every finite pattern as a subpattern, and choose n such that $G^n(t)$ is the quiescent configuration. It is easy to see that with this choice of n configuration $G^n(c)$ is quiescent for all $c \in S^{\mathbb{Z}^d}$.

2D nilpotency
Instance: A two-dimensional cellular automaton G
Problem: Is G nilpotent ?

Proposition 42 *Decision problem* 2D nilpotency *is undecidable. It is semi-decidable while the complement is not semi-decidable.*

Proof. For semi-decidability notice that, for $n = 1, 2, 3 \ldots$, we can effectively construct a cellular automaton whose global function is G^n and check whether the local rule of the CA maps everything into the quiescent state. If that happens for some n then we halt and report that the CA is nilpotent.

To prove undecidability we reduce the Tiling problem. For any given Wang tile set T we construct a cellular automaton whose state set is $S = T \cup \{q\}$ and the local rule turns a cell into state q except if the cell and its four neighbors are tiles that match in color, in which case the state is not changed. (This same construction was already used in Example 17). Let us prove that the CA is not nilpotent if and only if T admits a valid tiling.

"$\Longleftarrow$": Suppose a valid tiling exists. This tiling, as a configuration of the CA, is a fixed point so it never becomes quiescent. The CA is not nilpotent.

"$\Longrightarrow$": Suppose no valid tiling exists. Then there is number n such that no valid tiling of an $n \times n$ square exists. This means that after the first application of the CA, regardless of the initial configuration, state q appears in every $n \times n$ square. Since state q spreads it is clear that the configuration becomes eventually quiescent. Hence the CA is nilpotent. □

In fact, using the undecidability of the NW-deterministic tiling problem, we can prove that the nilpotency problem is undecidable even for one-dimensional CA:

1D nilpotency
Instance: A one-dimensional cellular automaton G
Problem: Is G nilpotent ?

Proposition 43 *Decision problem* 1D nilpotency *is undecidable. It is semi-decidable while the complement is not semi-decidable.*

Proof. Semi-decidability follows in the same way as in the two-dimensional case. To prove undecidability we reduce the NW-deterministic tiling problem. Let T be a given NW-deterministic tile set. We construct a one-dimensional CA whose state set is $S = T \cup \{q\}$ and the local rule turns a cell into state q except in the case that the cell and its right neighbor are in states $x, y \in T$, respectively, and tile $z \in T$ exists so that tiles x, y, z match as in Figure 24(a). In this case z is the new state of the cell. This same construction was already used in Example 18.

Let us prove that the CA is not nilpotent if and only if T admits a valid tiling.

"$\Longleftarrow$": Suppose a valid tiling exists. If $c \in T^{\mathbb{Z}}$ is a diagonal of this tiling then the configurations $G^n(c)$ in its orbit are subsequent diagonals of the same tiling, for all $n = 1, 2, \ldots$. This means that c never becomes quiescent, and the CA is not nilpotent.

"$\Longrightarrow$": Suppose no valid tiling exists. Then there is number n such that no valid tiling of an $n \times n$ square exists. This means that for every initial configuration $c \in S^{\mathbb{Z}}$ the configuration $G^{2n}(c)$ is quiescent: If it is not quiescent then a valid tiling of an $n \times n$ square can be read from the space time diagram of configurations $c, G(c), \ldots, G^{2n}(c)$. We conclude that the CA is nilpotent. □

Consider then the problem of determining if a given two-dimensional CA is surjective. To prove that this question is undecidable we reduce the Finite tiling problem. In the reduction we need a particular set D of 23 Wang tiles shown in Figure 30. The topmost tile is called blank. All other tiles have a unique incoming and outgoing arrow. In valid tilings arrows and labels must match. The non-blank tiles are considered directed: the follower of a tile is the neighbor directed to by the outgoing arrow on the tile. Since each non-blank tile has exactly one incoming arrow, it is clear that if the tiling is valid at a tile then the tile is the follower of exactly one of its four neighbors.

The tile at the center in Figure 30 where the dark and light thick horizontal lines meet is called the *cross*. It has a special role in the forthcoming proof. A *rectangular loop* is a valid tiling of a rectangle using tiles in D where the follower path forms a loop that visits every tile of the rectangle, and the outside border of the rectangle is colored blank. See Figure 31 for an example of a rectangular loop through a rectangle of size 12×7. (The edge labels are not shown for the sake of clarity of the figure.) It is easy to see that a rectangular loop of size $2n \times m$ exist for all $n \geq 2$ and $m \geq 3$. Any tile in an even column in the interior of the rectangle can be made to contain the unique cross of the rectangular loop.

The tile set D has a property that resembles the plane-filling property:

Lemma 44 *Let $t \in D^{\mathbb{Z}^2}$ be a tiling, and $\vec{p}_1, \vec{p}_2, \vec{p}_3, \ldots$ a path in t such that the tiling t is valid at $\vec{p}_i$ for all $i = 1, 2, 3, \ldots$. If the path covers only a finite number of different cells then the cells on the path form a rectangular loop.*

Proof. Because the path only covers a finite number of different cells, the path must form a cycle. A cycle must contain arrows in all four directions. In particular, an arrow to the left must appear. Only the tiles at the bottom row of Figure 30 involve left arrows, so the path must contain a horizontal segment of left arrows that ends in a lower left corner tile where the path turns upwards and has label SW. From there on the label SW forces the path to continue upwards (forming the left border of the rectangle) until it turns to the right, with label A. In this vertical segment the label of the arrow must change from SW to NW, so a unique horizontal dark thick line gets initiated. Label A forces the path to immediately turn down. The path can only continue downwards until it eventually turns when the tile below has label C. The only tile with label C on the top edge contains a left arrow, so the tile must be next to the lower left corner tile.

Next the path has no other choice but to turn up and to continue upwards until it reaches the top of the rectangle where label B on its left side forces it to turn right. Continuation of the same reasoning shows that the path must continue up and down filling the rectangle until at some point the downward path has labels NE. This label must be changed to SE at

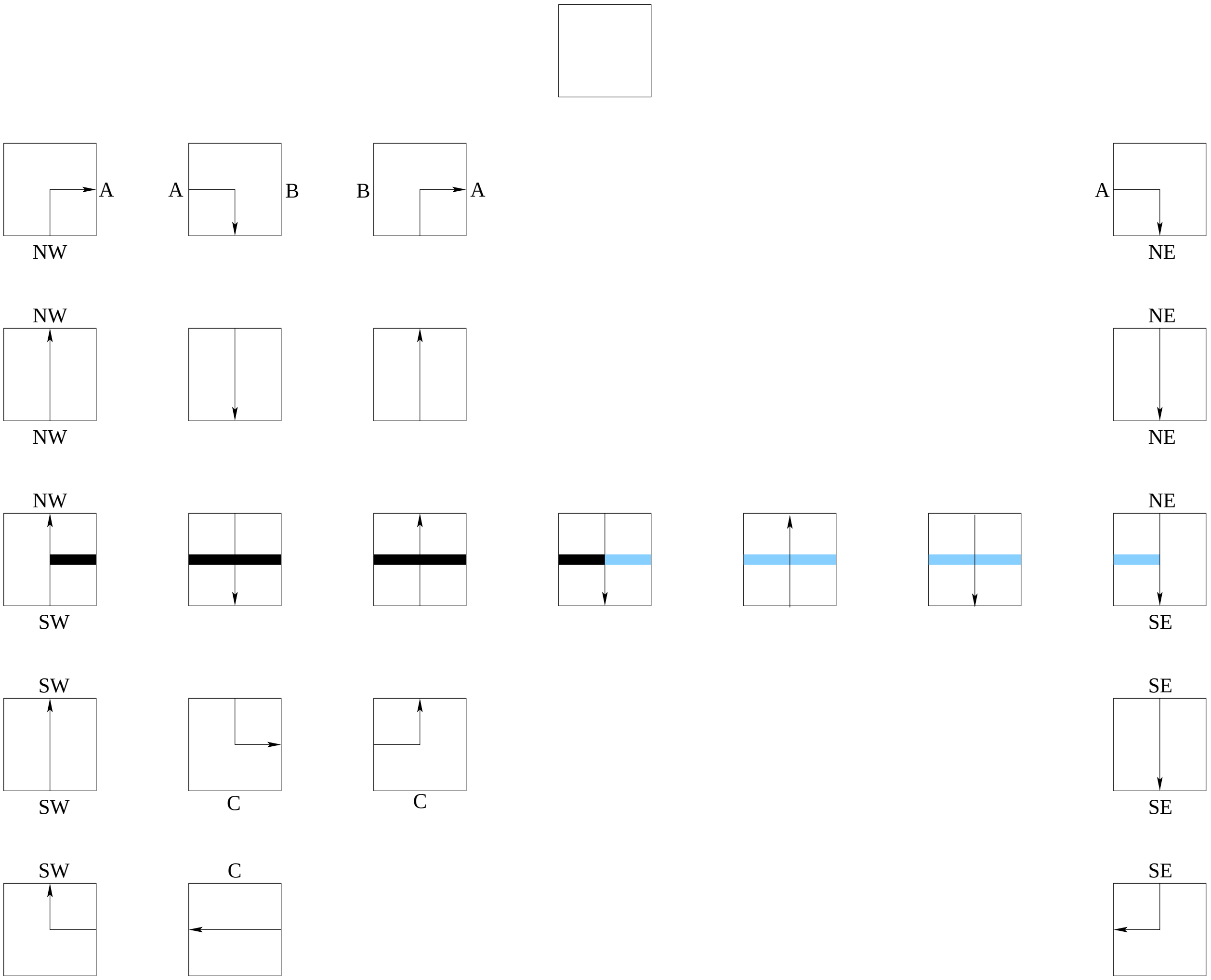


Figure 30: Tiles used in the proof of the undecidability of 2D surjectivity.

a tile where a horizontal light thick line ends. At the bottom of the rectangle the path must then turn leftwards to complete the cycle into a rectangular loop. The rectangle contains a unique horizontal thick bar, and the bar has to switch color once from dark to light, so the interior of the rectangle contains a unique cross. □

Now we are ready to prove the undecidability of 2D Surjectivity.

Proposition 45 *The decision problem* 2D Surjectivity *is undecidable. It is not even semi-decidable while the complement problem is semi-decidable.*

Proof. The semi-decidability of non-surjectivity was already discussed in Example 20, so it is enough to prove undecidability. We reduce the Finite tiling problem into 2D Surjectivity,

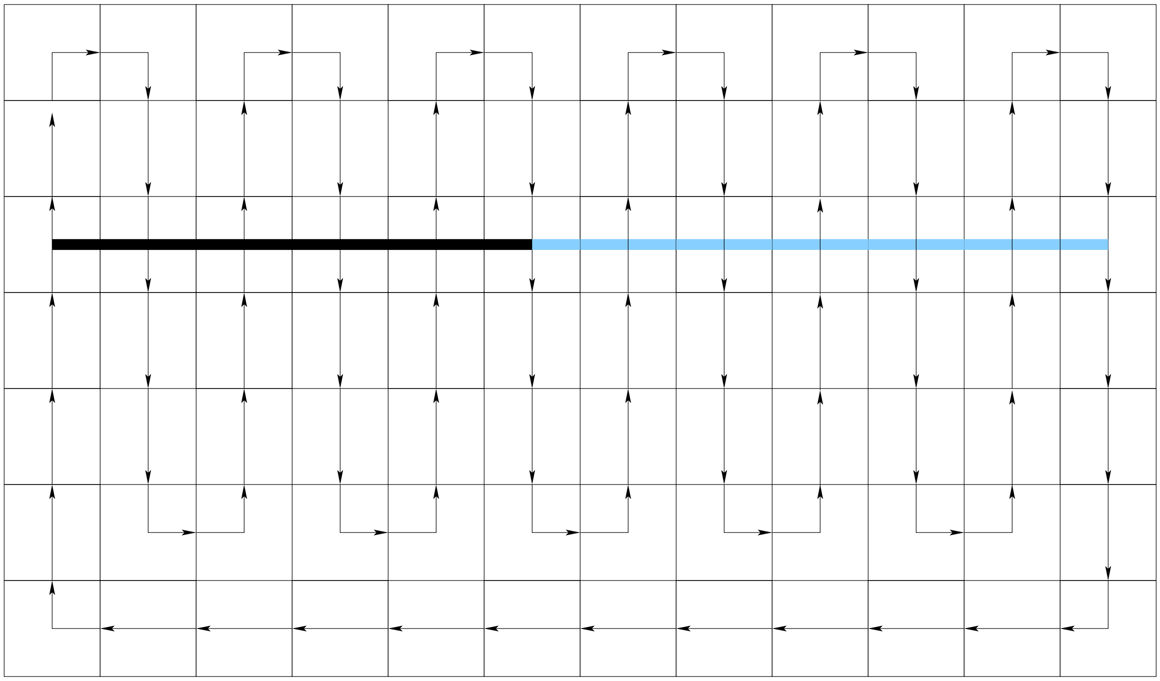


Figure 31: A rectangular loop of size 12 × 7.

using the tile set D discussed above in Lemma 44. Let b and c be the blank and the cross of set D. For any given tile set T with blank tile B we construct the following two-dimensional cellular automaton. The state set S contains triplets

$$(d, t, x) \in D \times T \times \{0, 1\}$$

under the following constraints:

- If $d = c$ then $t \neq B$, and
- if $d = b$ or d is any tile containing label SW, SE, NW, NE, A, B or C, then $t = B$.

In other words, the cross must be associated with a non-blank tile in T while the blank of D as well as all the tiles on the boundary of a rectangular loop are forced to be associated with the blank tile of T. The triplet $(b, B, 0)$ where both tile components are blank and the bit is 0 is the quiescent state of the CA. The local rule is as follows: Let (d, t, x) be the current state of a cell.

- If $d = b$ then the state is not changed.
- If $d \neq b$ then the cell verifies the validity of the tilings according to both D and T at the cell. If either tile component has a tiling error then the state is not changed. If both tilings are valid then the cell modifies its bit component by adding the bit of its follower modulo 2.

Let us prove that this CA is not surjective if and only if T admits a valid, finite, non-trivial tiling.

"$\Longleftarrow$": Suppose a valid, finite, non-trivial tiling $t \in T^{\mathbb{Z}^2}$ exists. Consider a configuration of the CA whose T-components form the valid tiling t and the D-components form a rectangular loop whose interior covers all non-blank elements of t. Tiles outside the rectangle are all blank and have bit 0. The cross can be positioned so that it is in the same cell as some non-blank tile in t. In such a configuration both T and D tilings are everywhere valid. The CA updates the bits of all tiles in the rectangular loop by performing modulo 2 addi tion with their followers, while the bits outside the rectangle remain 0. We get two different configurations that have the same image: In c_0 all bits in the rectangle are equal to 0 while in c_1 they are all equal to 1. The local rule updates the bits so that $G(c_0) = G(c_1) = c_0$. Configurations c_0 and c_1 only differ in a finite number of cells, so it follows from the Garden-of-Eden theorem (Proposition 18) that G is not surjective.

"$\Longrightarrow$": Suppose then that the CA is not surjective. According to the Garden-of-Eden theorem (Proposition 16) there are two different finite configurations c_0 and c_1 such that $G(c_0) = G(c_1)$. Since only bit components of states are changed, the tilings in c_0 and c_1 according to D- and T-components of the states are identical. There is a cell $\vec{p}_1$ such that c_0 and c_1 have different bits at cell $\vec{p}_1$. Since these bits become identical in the next configuration, the D-tiling must be correct at $\vec{p}_1$ and c_0 and c_1 must have different bits in the follower position $\vec{p}_2$. We repeat the reasoning and obtain an infinite sequence of positions $\vec{p}_1, \vec{p}_2, \vec{p}_3, \ldots$ such that each $\vec{p}_{i+1}$ is the follower of $\vec{p}_i$, and the D tiling is correct at each $\vec{p}_i$. Moreover, c_0 and c_1 have different bits in each position $\vec{p}_i$. Because configurations c_0 and c_1 are finite we see that the path can only contain a finite number of distinct cells. It follows then from Lemma 44 that the path must form a valid rectangular loop.

Also the tiling according to the T-components must be valid at each cell of the path. Because of the constraints on the allowed triplets, the T-components on the boundary of the rectangle are the blank B, while the cross in the interior contains a non-blank element of T. Hence there is a valid tiling of a rectangle according to T that contains a non-blank tile and has a blank boundary. This means that a finite, valid and non-trivial tiling is possible.

□

Finally, consider the problem of determining if a given two-dimensional CA is reversible. The proof of its undecidability is similar to the previous proof. Only, instead of the Finite tiling problem we use the Tiling problem, and the special set D if directed tiles is replaced by a tile set with the plane filling property.

Proposition 46 *Decision problem* 2D Reversibility *is undecidable. It is semi-decidable while the complement problem is not.*

Proof. The semi-decidability was already shown in Example 20. Let us prove 2D Reversibility undecidable by reducing the Tiling problem into it. In the reduction we use a set D of directed tiles that has the plane filling property. The existence of such D was stated (without proof) in Proposition 32.

Let T be a given set of Wang tiles that is an instance of the Tiling problem. One can effectively construct a two-dimensional CA whose state set is

$$S = T \times D \times \{0, 1\}$$

and the local rule updates the bit component of a cell as follows:

- If either the T- or the D-components contain a tiling error at the cell then the state of the cell is not changed, but
- if the tilings according to both T- and D-components are valid at the cell then the bit of the follower cell (according to the direction in the D-component) is added to the present bit value modulo 2.

The tile components are not changed. Let us prove that this CA is not injective (and hence not reversible) if and only if T admits a valid tiling.

"$\Longleftarrow$": Suppose a valid tiling exists. Construct two configurations c_0 and c_1 where the T- and D-components form the same valid tilings $t \in T^{\mathbb{Z}^2}$ and $d \in D^{\mathbb{Z}^2}$, respectively. In c_0 all bits are 0 while in c_1 they are all 1. Since the tilings are everywhere valid, every cell performs modulo 2 addition of two bits, which means that every bit becomes 0. Hence $G(c_0) = G(c_1) = c_0$, and G is not injective.

"$\Longrightarrow$": Suppose then that the CA is not injective. There are two different configurations c_0 and c_1 such that $G(c_0) = G(c_1)$. Tile components are not modified by the CA so they are identical in c_0 and c_1. There is a cell $\vec{p}_1$ such that c_0 and c_1 have different bits at cell $\vec{p}_1$. Since these bits become identical in the next configuration, the D-tiling must be correct at $\vec{p}_1$ and c_0 and c_1 must have different bits in the follower position $\vec{p}_2$. We repeat the reasoning and obtain an infinite sequence of positions $\vec{p}_1, \vec{p}_2, \vec{p}_3, \ldots$ such that each $\vec{p}_{i+1}$ is the follower of $\vec{p}_i$, and the D tiling is correct at each $\vec{p}_i$. It follows from the plane filling property of D that path $\vec{p}_1, \vec{p}_2, \vec{p}_3, \ldots$ covers arbitrarily large squares. Also the tiling according to the T-components must be valid at each cell of the path. Hence tile set T admits valid tilings of arbitrarily large squares, and therefore it admits a valid tiling of the entire plane.

□

Note: The two-dimensional cellular automata that we constructed in the proofs of Propositions 42, 45 and 46 use the von Neumann neighborhood. So the decision problems 2D Nilpotency, 2D Surjectivity and 2D Reversibility are undecidable even when the input instance is restricted to CA with the von Neumann neighborhood.

The one-dimensional CA constructed in the proof of Proposition 43 has radius-$\frac{1}{2}$ neighborhood. Also, the quiescent state q is spreading: Any cell whose neighborhood contains state q turns to state q. So the decision problem 1D Nilpotency is undecidable among radius-$\frac{1}{2}$ CA that have a spreading state.

Another fact to observe is that all undecidability results of this section apply to d-dimensional CA for any $d \geq 2$, even though we only proved them in the case $d = 2$. Namely, for any $d > 2$ a given two-dimensional CA A can be converted into a d-dimensional CA consisting of a $(d-2)$-dimensional grid of independent two-dimensional layers, each of which operates as A. This d-dimensional CA is nilpotent, surjective or reversible if and only if A is nilpotent, surjective or reversible, respectively.

We finish this section by defining busy beaver -like, very rapidly growing functions associated with the undecidable decision problems above.

Example 24. Let us define following functions $n, s, r : \mathbb{Z}_+ \longrightarrow \mathbb{Z}_+$: For every $k \in \mathbb{Z}_+$

- $n(k)$ is the largest t such that there is a nilpotent, one-dimensional, radius-$\frac{1}{2}$ cellular automaton G with k states, and a configuration c such that $G^t(c)$ is not the quiescent configuration.

- $s(k)$ is largest t such that there is a non-surjective, two-dimensional CA that uses the von Neumann neighborhood and has k states, such that there is no orphan pattern of size $t \times t$.

- $r(k)$ is the largest t such that there is a reversible, two-dimensional CA G with k states and the von Neumann neighborhood such that the inverse of G is not obtained using the radius-t neighborhood.

Each of the functions is well defined since its value for any k is the maximum among a finite number of positive integers. Yet no algorithm can compute an upper bound for any of the functions. If some algorithm could produce for every given k a number t such that $n(k) \leq t$, $s(k) \leq t$ or $r(k) \leq t$ then this algorithm could be used to solve the decision problem 1D Nilpotency, 2D Surjectivity or 2D Reversibility, respectively. In each case such value t would namely provide a bound on the size of instances that the corresponding semi-algorithm needs to check.

We conclude that there are nilpotent CA that have configurations that survive very long time before becoming quiescent, there are non-surjective CA whose smallest orphan is very large, and there are reversible CA whose inverse CA have very large neighborhoods. □

3.5 Computational universality

General purpose computers are computationally universal: they can simulate any semi-algorithm if the semi-algorithm is encoded properly and is included as part of the input. Note how important it is for us that computational universality is at all possible – otherwise we would have to build a new computer for each computational task!

A recursively enumerable language $L \subseteq \Sigma^*$ is *r.e.-complete* if every recursively enumerable language is many-one reducible to L. In other words, for every r.e. language $K \subseteq \Delta^*$ there exists an algorithm that takes as input an arbitrary word $u \in \Delta^*$ and outputs a word $v \in \Sigma^*$ with the property that $u \in K$ if and only if $v \in L$. In terms of decision problems, we call a semi-decidable problem U r.e.-complete if for every semi-decidable problem P there exists an algorithm that converts instances of P into equivalent instances of U, that is, positive and negative instances of P get converted to positive and negative instances of U, respectively.

Semi-algorithms for r.e.-complete decision problems are called *computationally universal*. This means that a semi-algorithm A is universal iff for every semi-algorithm B there exists a conversion algorithm that converts an arbitrary input of B into an equivalent input of A.

Proposition 47 *Decision problem* Semi-algorithm halting *is r.e.-complete.*

Proof. Let P be an arbitrary semi-decidable decision problem. There is a semi-algorithm A for P. An arbitrary instance w of P can be trivially converted into the equivalent instance $\langle A \rangle, w$ of Semi-algorithm halting.

□

Note that if an r.e.-complete decision problem P can be many-one reduced to a semi-decidable decision problem Q, then Q is also r.e.-complete. This follows from the fact that many-one reducibility $\leq_m$ is a transitive property: if $R \leq_m P$ and $P \leq_m Q$ then $R \leq_m Q$. Because all undecidability proofs that we made for tilings and cellular automata (and also the proofs that were skipped) were obtained by a chain of many-one reductions from Semi-algorithm halting we obtain from Proposition 47 the following:

Corollary 48 *The following decision problems are r.e.-complete:* TM halting on blank tape*,* Finite tiling problem*,* Periodic tiling problem*,* 1D nilpotency *and* 2D Reversibility*. The complements of the following are r.e.-complete:* Tiling problem with the seed tile*,* Tiling problem*,* NW-deterministic tiling problem *and* 2D Surjectivity*.* □

Any semi-algorithm for any of the problems in the corollary is computationally universal.

A Turing machine will be called computationally universal if the language it recognizes is r.e.-complete. Such machines exist because r.e.-complete languages exist, and every such language is recognized by some Turing machine.

The goal of this section is to investigate computational universality in cellular automata. Language recognition by Turing machines is well defined using blank tape symbols and initial and accepting states, so the definition of computational universality is also precise. In the case of cellular automata no such widely accepted standard definition for universality exists. Various ways to encode the input instance, and different acceptance conditions can be found in the literature. A precise definition would be needed if we wanted to prove that some CA are not computationally universal. Our interest, however, is to prove that some CA are computationally universal, and for that purpose we simply choose the encoding format and acceptance condition suitably for that particular CA, and say that the CA is universal under these conditions.

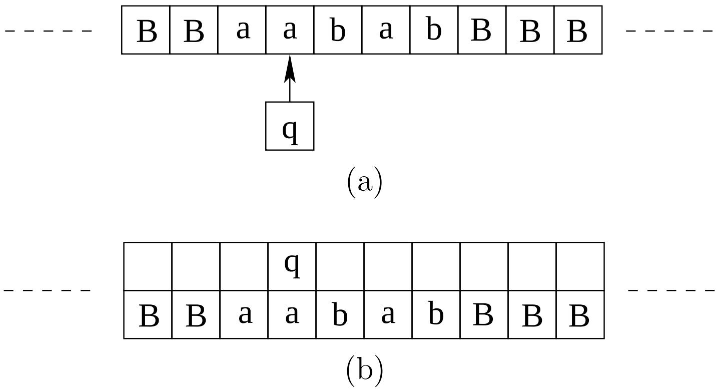
(Observe, however, that even in the case of Turing machines alternative notions of universality exist. For example, one could define universality by the condition that the problem whether a given finite configuration of the Turing machine evolves into another given finite configuration is r.e. complete. By finite configuration we mean a configuration where only finitely many cells contain a non-blank tape symbol. Another definition used in the literature calls a Turing machine universal if the problem whether a given finite configuration evolves into an accepting configuration is r.e. complete.)

Let us first show an obvious way how an arbitrary Turing machine can be simulated by a CA. The following construction simply implements a Turing machine M in one-dimensional CA. Configurations of the CA consist of two tracks: one track stores the tape content while the other track has just one non-quiescent cell that contains the Turing machine state. Each Turing machine move is implemented by the CA local rule so that changes occur only in two cells: the cells containing the Turing machine before and after the move.

Let Q and Γ be the state set and tape alphabet of Turing machine M, and let δ be its transition function. The state set S of the CA consists of pairs

$$(q, a) \in (Q \cup \{0\}) \times \Gamma.$$

where $0 \notin Q$ is used to indicate that the Turing machine is not present in this cell. Pairs whose first component is $q \neq 0$ indicate that the Turing machine head is reading the cell in state q.



(a)

(b)

Figure 32: (a) Turing machine configuration converted into (b) CA configuration.

The CA uses radius-1 neighborhood and the local rule is defined as follows: If the neighborhood of a cell contains no Turing machine head then the state of the cell is not changed. If the neighborhood contains more than one Turing machine heads then the local rule can be defined arbitrarily, e.g., no action is taken. Such situation will never happen in valid simulations of the Turing machine. Suppose then that exactly one neighbor of the cell contains the Turing machine head. A change in the state of the cell occurs only in the following cases, where $q \neq 0$:

- The cell itself is in state (q, a). Let $\delta(q, a) = (q', a', d)$. The new state of the cell is $(0, a')$.
- The right neighbor is in state (q, a) and $\delta(q, a) = (q', a', -1)$. Then the new state of the cell will be (q', x) where $(0, x)$ is its old state.
- The left neighbor is in state (q, a) and $\delta(q, a) = (q', a', +1)$. Then the new state of the cell will be (q', x) where $(0, x)$ is its old state.

The quiescent state of the CA is the pair $(0, B)$ where B is the blank tape symbol of the Turing machine.

Let us encode Turing machine configurations as CA configurations as follows: For all $q \in Q$, $i \in \mathbb{Z}$ and $t \in \Gamma^{\mathbb{Z}}$ the Turing machine configuration (q, i, t) is encoded as the CA configuration $c \in S^{\mathbb{Z}}$ where for every $j \in \mathbb{Z}$

$$c(j) = \begin{cases} (q, t(j)) & \text{if } j = i, \\ (0, t(j)) & \text{if } j \neq i. \end{cases}$$

Let us denote then

$$c = E(q, i, t)$$

and call

$$E : Q \times \mathbb{Z} \times \Gamma^{\mathbb{Z}} \longrightarrow S^{\mathbb{Z}}$$

the encoding function. See Figure 32 for an illustration for the encoding of a configuration.

It is clear from the definition of the CA that the diagram

$$\begin{tikzcd}
Q \times \mathbb{Z} \times \Gamma^{\mathbb{Z}} \arrow[r, "\vdash"] \arrow[d, "E"] & Q \times \mathbb{Z} \times \Gamma^{\mathbb{Z}} \arrow[d, "E"] \\
S^{\mathbb{Z}} \arrow[r, "G"] & S^{\mathbb{Z}}
\end{tikzcd}$$

commutes. In other words, if the Turing machine changes (q, i, t) into (q', i', t') in k steps then the CA changes configuration $E(q, i, t)$ into $E(q', i', t')$ in k steps. This means that the CA simulates the Turing machine moves. If the Turing machine M that the construction was performed on is computationally universal then the resulting CA can be called computationally universal as well. In this form of universality the input word is encoded as a finite initial configuration, and the word is accepted if a configuration with some cell in state (q_a, x) for some $x \in \Gamma$ is eventually reached, where q_a is the accepting state of the Turing machine.

Proposition 49 *There is a one-dimensional CA and a subset $F \subseteq S$ of states such that the following decision problem is r.e. complete: "Does a given finite configuration c evolve into a configuration where some cell is in state belonging to F ?"*

Proof. Clearly the given decision problem is semi-decidable for any CA. The completeness is proved by performing the construction above for a Turing machine M that recognizes an r.e. complete language $L \subseteq \Sigma^*$ and by choosing

$$F = \{(q_a, x) \mid x \in \Gamma\}.$$

Then for any $w \in \Sigma^*$ we have $w \in L$ if and only if the finite configuration $c = E(q_0, 1, t_w)$ is a positive instance of the given decision problem, where q_0 is the initial state of the Turing machine, and $t_w \in \Gamma^{\mathbb{Z}}$ has word w written in positions $1, 2, \ldots, |w|$ and blank B in all other cells. □

It is an easy matter to modify the CA for alternative forms of universality. The following proposition lists a few possibilities:

Proposition 50 *For each of the following decision problems there is a one-dimensional CA such that the given decision problem is r.e. complete:*

(a) Does a given finite configuration c evolve into the quiescent configuration ?

(b) For a fixed spreading state s, does a given finite configuration c evolve into a configuration e in which state s appears ?

(c) Does a given finite configuration c evolve into a fixed point ?

(d) Given two finite configurations c and e, does c evolve into e, that is, does there exist $n \geq 0$ such that $G^n(c) = e$?

Proof. First note that all questions (a)–(d) are semi-decidable, so it is enough to show completeness.

Consider a universal Turing machine that has the property that it never writes the blank symbol B. Such Turing machine is easily obtained from any universal machine by introducing a new symbol B' that behaves exactly as B, and the Turing machine always writes B' on the tape instead of B. This modification is done so that we can easily recognize the active part of the tape.

Let us modify the previous construction of the CA A that simulates a universal Turing machine:

For (a) we modify A it so that if an accepting state in F appears, it immediately sends two signals, one to the left and another one to the right, that change all states into the quiescent state. When the signal reaches the quiescent state (indicating that the active part of the tape ends) the signal disappears.

For (b) we modify A so that we introduce new spreading state s, and modify the local rule so that states in F become s, regardless of their neighbors.

For (c) we introduce a new inactive state s. Cell in state s remains s and also all its neighbors do not change their states. States in F become this s in one step. Then accepting configuration is a fixed point, while non-accepting configurations are not fixed points since the Turing machine always moves left or right.

For (d) we use the same construction as in (a) and choose e always to be the quiescent configuration. □

4 Reversible cellular automata

This section takes an in-depth look into reversible CA (RCA). As we have seen previously, reversible CA are exactly the injective ones. In two- and higher dimensional spaces it is undecidable if a given CA is reversible, which implies that the inverse automaton can have a very large neighborhood: no computable upper bound can be given.

We start this section by providing two methods for constructing reversible CA.

4.1 Partitioned CA

Partitioned CA (PCA) are particular types of reversible CA. The state set of a PCA with m neighbors is the cartesian product

$$S = S_1 \times S_2 \times \ldots \times S_m$$

of m finite sets $S_1, S_2, \ldots, S_m$. Elements $s_i \in S_i$ of state $s = (s_1, s_2, \ldots, s_m)$ are called the components of s. The i'th components of the states of all cells constitute the i'th *track* of a configuration.

The local rule is given by a permutation

$$\pi : S \longrightarrow S$$

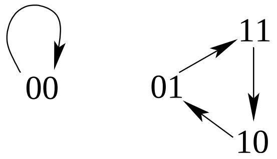
of the state set. Each cell takes the i'th component of its i'th neighbor, merges these for all $i = 1, 2, \ldots, m$ together into an element of S, and applies permutation π to the result. More precisely, by denoting by s_i^j the i'th component of the j'th neighbor, we get the local rule

$$f[(s_1^1, s_2^1, \ldots, s_m^1), (s_1^2, s_2^2, \ldots, s_m^2), \ldots, (s_1^m, s_2^m, \ldots, s_m^m)] = \pi(s_1^1, s_2^2, \ldots, s_m^m).$$

Example 25. Let $d = 1$ and $N = (0, 1)$ so $m = 2$. Let

$$S = \{0, 1\} \times \{0, 1\} = \{00, 01, 10, 11\}.$$

There are $4! = 24$ different permutations of S. Take, for example, permutation π that maps



Then, for example, initial configuration where one cell is in state 11 while all others are in state 00 evolves as shown in Figure 33. In the illustration white and black color are used for 0 and 1, respectively, and the second component is drawn under the first component in each cell.

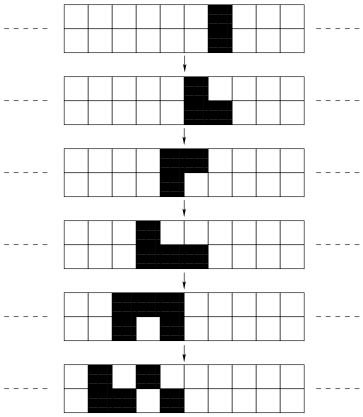


Figure 33: Beginning of an orbit in a partitioned CA.

One step of the PCA consists of two simple CA operations: (1) the second track is shifted one position to the left, and (2) permutation π is applied at each cell. Since both operations (1) and (2) are clearly one-to-one, the composite PCA is reversible. The inverse function first applies the inverse permutation π^{-1} at each cell and then shifts the second components one position to the right. □

Analogously to the example, any PCA is a composition of two obviously reversible CA functions:

(1) For every $i = 1, 2, \ldots, m$, apply translation $\tau_{\vec{n}_i}$ on the i'th track, where $\vec{n}_i$ is the i'th element of the neighborhood vector.

(2) Apply permutation π at every cell.

So we have the following result:

Proposition 51 *Partitioned CA are reversible.* □

Despite their apparent simplicity, PCA can have complex behavior. In fact, they can exhibit computational universality.

Proposition 52 *There is a one-dimensional PCA and a subset $F \subseteq S$ of states such that the following decision problem is r.e. complete: "Does a given finite configuration c evolve into a configuration where some cell is in state belonging to F ?"*

Proof. Construction is similar in flavor to our earlier universal CA. However, reversibility means that no information may be erased. For this reason we introduce a new "garbage" track where information about previous states of the Turing machine are stored and shifted away to the left. We shift the garbage away quickly, two cells in one step, to make sure that the garbage from previous moves does not interfere with forthcoming moves of the Turing machine.

The track that contains the Turing machine state will be split into two tracks where the first track is used when the machine moves left and the second track is used when it moves right. This is done to allow simple partitioning of the states.

Let us start the construction based on a universal Turing machine M. As always, we denote by Γ and Q the tape alphabet and the state set of M, respectively, and by δ the transition function. The state set of the corresponding PCA is

$$S = S_1 \times S_2 \times S_3 \times S_4$$

where

$$\begin{array}{rcl} S_1 & = & \Gamma, \\ S_2 & = & Q \cup \{0\}, \\ S_3 & = & Q \cup \{0\}, \text{ and} \\ S_4 & = & (Q \times \Gamma \times \{L, R\}) \cup \{0\}. \end{array}$$

Here, 0 is a new symbol. The neighborhood vector is $(0, 1, -1, 2)$. The order of the elements in the vector matters since they correspond to the four tracks. In other words, the translation step of the PCA consists of shifting the second and third track that contain the Turing machine state information one cell two the left and right, respectively, and the fourth garbage track is shifted two positions to the left.

We first define a partial function $\pi : S \longrightarrow S$, and then we complete it into a full permutation of S. For every $a \in \Gamma$ and $q \in Q$ partial function π maps

$$\begin{array}{lcll} (a, q, 0, 0) & \mapsto & (a', q', 0, (q, a, L)) & \text{if } \delta(q, a) = (q', a', -1), \\ (a, q, 0, 0) & \mapsto & (a', 0, q', (q, a, L)) & \text{if } \delta(q, a) = (q', a', +1), \\ (a, 0, q, 0) & \mapsto & (a', q', 0, (q, a, R)) & \text{if } \delta(q, a) = (q', a', -1), \text{ and} \\ (a, 0, q, 0) & \mapsto & (a', 0, q', (q, a, R)) & \text{if } \delta(q, a) = (q', a', +1). \end{array}$$

Notice that (i) the information about the first three components is stored in the last track, and (ii) the new state q' is stored on the second or the third track depending on whether the machine moves left or right. From there it is moved in the next translation phase into the correct cell. In addition, to keep the inactive part of the TM tape unchanged we map

$$(a, 0, 0, g) \mapsto (a, 0, 0, g)$$

for all $a \in S_1$ and $g \in S_4$. These partially defined values of π are the only ones that will be used by any cell in a valid simulation of the Turing machine. The simulation begins with the initial configuration where the input word w is written in cells $1, 2, \ldots, |w|$ of the first track, and tracks 2,3 and 4 contain 0 everywhere except that the third component of cell 0 contains the initial state of the Turing machine. See Figure 34 for the initial configuration with input word $w = baab$, and for the first two steps under the assumption that $\delta(q_0, b) = (q, x, -1)$ and $\delta(q, b) = (r, a, R)$.

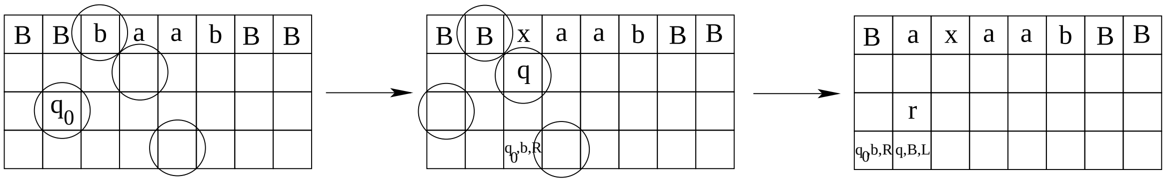


Figure 34: Sample first steps by a universal PCA.

It is clear that the first three tracks properly simulate the Turing machine, step-by-step. The garbage track is shifted left by two cells, which makes sure that an empty garbage slot is always available at the active cell. It follows that the input word is accepted if and only if the accepting state q_a eventually appears on the second or the third track.

The function π was only partially defined, but observe that it is one-to-one. Hence one can (arbitrarily) complete it into a permutation of S. □

Next we show that all one-dimensional reversible CA are, in some sense, partitioned CA. First we define a blocking function B_m that combines segments of m consecutive cells into a single "super cell". Consider state set S. For every positive integer m function

$$B_m : S^{\mathbb{Z}} \longrightarrow (S^m)^{\mathbb{Z}}$$

is defined as follows: For every $c \in S^{\mathbb{Z}}$ we have $B_m(c) = e$ where for every $i \in \mathbb{Z}$

$$e(i) = (c_{mi+1}, c_{mi+2}, \ldots, c_{mi+m}) \in S^m.$$

See Figure 35 for an illustration of B_3. Function B_m is a bijection so it has the inverse function B_m^{-1} that breaks the super cells back to their components.
If $G : S^{\mathbb{Z}} \longrightarrow S^{\mathbb{Z}}$ is a CA function over state set S then

$$B_m^{-1} \circ G \circ B_m : (S^m)^{\mathbb{Z}} \longrightarrow (S^m)^{\mathbb{Z}}$$

is also a CA function, the *m-block presentation* of G. The state set of of the m-block presentation is S^m.

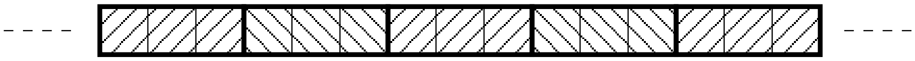


Figure 35: Blocking function B_3 merges segments of three cells into "super cells"..

Proposition 53 *For every one-dimensional reversible CA G there is a positive integer n such that the $2n$-block presentation of $G \circ \sigma^n$ is a radius-$\frac{1}{2}$ partitioned CA.*

Note: When we state that a CA A is a PCA we mean that it is isomorphic to a PCA, that is, A becomes partitioned by renaming the states. More precisely, there is bijection from the state set of A onto the state set of a partitioned CA B that commutes with the CA evolutions according to A and B.

Proof. In the proof we frequently use the following notation: If c is a configuration and i, j are integers, $i \leq j$, then

$$c_{[i,j]} = c_i c_{i+1} \ldots c_j.$$

In other words, $c_{[i,j]}$ is the pattern that appears in c in the segment $i, i+1, \ldots, j$ of cells.

Let r be a positive number such that both G and G^{-1} are defined by radius-r cellular automata. Let $n = 3r$.

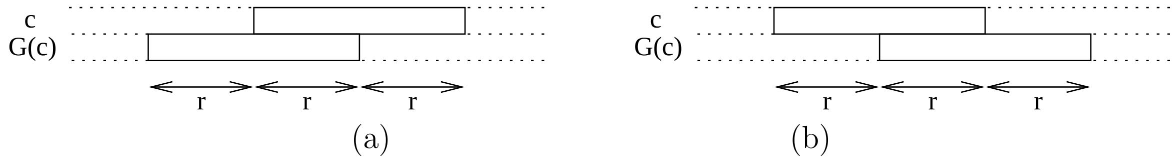


Figure 36: Extracting (a) right stairs, and (b) left stairs from configuration c.

Right stairs are pairs of patterns of length $2r$ extracted from configurations c and $G(c)$ in a staggered way, as indicated in Figure 36(a). Precisely, the set of right stairs is

$$\mathcal{R} = \{(c_{[0,2r-1]}, G(c)_{[-r,r-1]}) \mid c \in S^{\mathbb{Z}}\} \ \subseteq S^{2r} \times S^{2r}.$$

Analogously, left stairs are extracted as shown in Figure 36(b). The set of left stairs is

$$\mathcal{L} = \{(G(c)_{[0,2r-1]}, c_{[-r,r-1]}) \mid c \in S^{\mathbb{Z}}\} \ \subseteq S^{2r} \times S^{2r}.$$

Notice that any right stair (extracted from some $c \in S^{\mathbb{Z}}$) and any left stair (extracted from some $e \in S^{\mathbb{Z}}$) can in fact be extracted from the same configuration back-to-back, see Figure 37. This follows from the fact that the three parts of stairs are at least as long as the radius of the CA. The states in c to the right of the right stair do not affect the stair, and analogously the states in e to the left of the left stair have no affect on the stair, so the stairs appear back-to-back in the configuration whose left and right halves come from c and e, respectively.

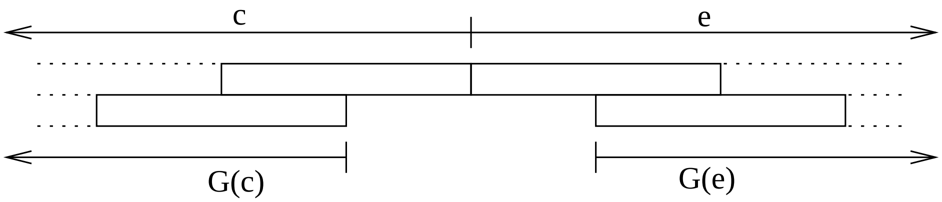


Figure 37: Arbitrary right and left stairs can be extracted back-to-back from the same configuration.

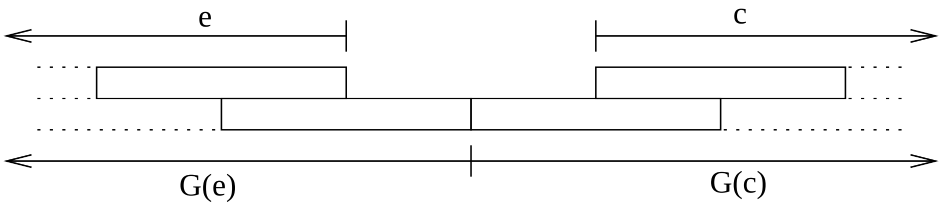


Figure 38: Extracting left and right stairs from the same configuration.

Analogously, using the local rule of G^{-1}, we see that the stairs can be extracted from the same configuration consecutively in the reverse order, as illustrated in Figure 38.

Let us show next that there is a natural bijection

$$\varphi : S^{6r} \longrightarrow \mathcal{R} \times \mathcal{L}$$

that maps for every $c \in S^{\mathbb{Z}}$

$$c_0 c_1 \ldots c_{6r-1} \mapsto \left[(c_{[4r,6r-1]}, G(c)_{[3r,5r-1]}), (G(c)_{[r,3r-1]}, c_{[0,2r-1]}) \right].$$

Figure 39 illustrates φ. In the following reasoning we refer to the shaded segment in the figure that is mapped by φ into the pair of striped stairs. Observe the following:

(a) Function φ is well defined, because the states in c that are outside the shaded segment has no influence on the two stairs.

(b) Function φ is one-to-one: Using the local rule of G^{-1} we see that the lower portions of the two striped stairs in Figure 39 uniquely determine the middle part of the shaded segment.

(c) Function φ is surjective: We know that any left and right stairs can be extracted back-to-back from the same configuration, so for any pair of stairs a shaded segment exists that is mapped by φ to the stairs.

We conclude from (a)–(c) that function φ is a bijection. This immediately implies that

$$|\mathcal{L}| \cdot |\mathcal{R}| = |S|^{6r}.$$

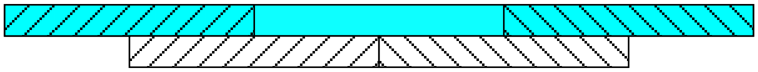


Figure 39: Bijection φ maps the shaded segment into the pair of striped stairs.

Analogously, we have a bijection

$$\psi : \mathcal{R} \times \mathcal{L} \longrightarrow S^{6r}$$

that maps a right stair/left stair pair into the $6r$ cells wide segment of $G(c)$, shown shaded in Figure 40.

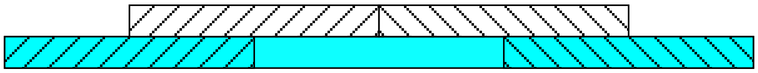


Figure 40: Bijection ψ maps the pair of striped stairs into the shaded segment.

Now we are ready to define our partitioned CA. It uses radius-$\frac{1}{2}$ neighborhood $N = (0, 1)$. The state set is the cartesian product $\mathcal{R} \times \mathcal{L}$. Following our notation for partitioned CA this means that in the state set $S_1 \times S_2$ we have $S_1 = \mathcal{R}$ and $S_2 = \mathcal{L}$. The permutation of the PCA is

$$\pi = \psi \circ \varphi.$$

This PCA is isomorphic to the $6r$-block presentation of $G \circ \sigma^{3r}$ where the isomorphism is given by renaming each $s \in S^{6r}$ as $\varphi(s)$. Indeed, referring to Figure 41, if we use the blocking function B_{6r} to partition a configuration $c \in S^{\mathbb{Z}}$ into segments $\ldots, s_1, s_2, \ldots$ of length $6r$, apply function φ on each segment, translate the left stairs of the blocks one segment to the left, apply function ψ, and finally break the segments using B_{6r}^{-1}, we clearly end up with configuration $\sigma^{3r}(G(c))$, as required. □

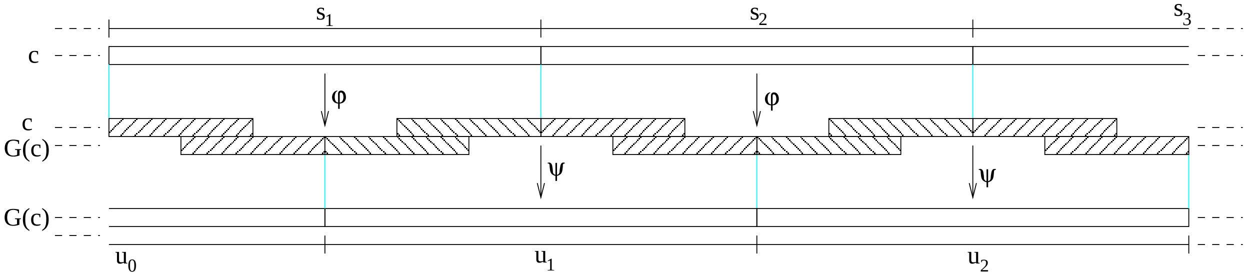


Figure 41: Presenting a one-dimensional reversible CA as a partitioned CA.

4.2 Margolus neighborhood

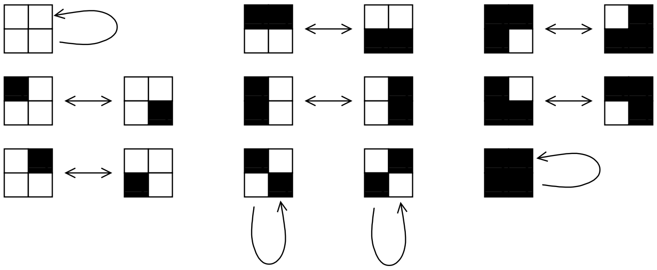
Using the so called *Margolus neighborhood* is a way to produce reversible CA. It can be viewed as a particular case of partitioned CA. Two well known two-dimensional examples that use this neighborhood are the *Billiard Ball CA* by Margolus and a lattice gas CA called *HPP*.

Consider a two-dimensional configuration $c \in S^{\mathbb{Z}^2}$. Partition it into two-by-two blocks, and apply a fixed bijection $\pi : S^4 \longrightarrow S^4$ on each block. Then change the partitioning by translating it one cell horizontally and vertically, and apply the same π again. Combined these two rounds define a reversible CA.

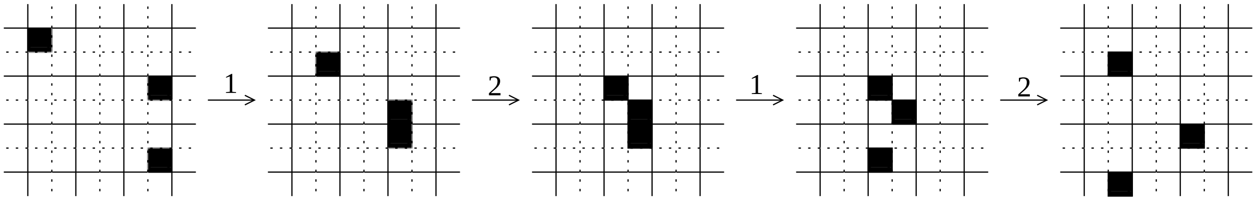
Let us start with a very simple example that uses two states

$$S = \left\{ \blacksquare, \square \right\}$$

and the following bijection π:



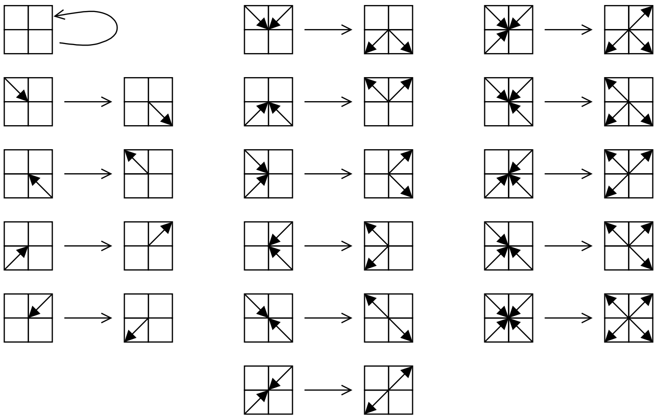
A configuration is updated in two rounds. The following illustration shows two iterations through both rounds:



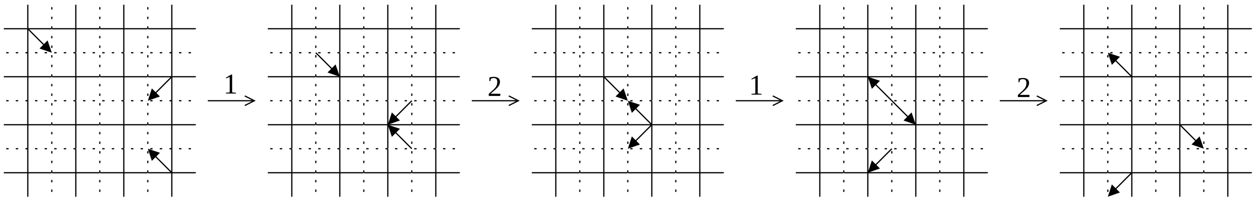
In both rounds the configuration is partitioned in two-by-two blocks of cells, and π is applied inside each block. In the first round the solid partitioning is used, while in the second round the dotted partitioning is used. Notice that this π is an involution (its own inverse), so the inverse CA uses the same rule, just that the dotted partitioning is used first, followed by the solid partitioning.

Strictly speaking, the position of a cell inside the two-by-two block affects its local rule, so there are four different local rules used at different positions. In this sense, in order to have a CA where all cells use the same local rule we have to consider the "super cells" formed by two-by-two blocks. So this CA, in fact, has $2^4 = 16$ states.

A good interpretation of this CA is to view each black state as a particle that moves diagonally on the plane. The position of the particle in the next two-by-two block to be updated determines the direction of the particle: It moves towards the center of the next update block. Under this interpretation, and keeping in mind that the two partitionings are alternated, the bijection π can be rewritten as

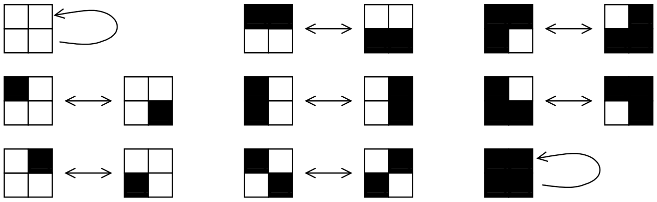


where we denote by an arrow the direction of each particle. Our sample iteration becomes

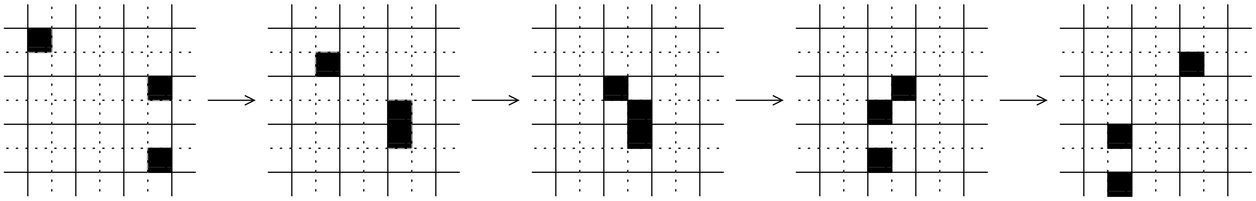


It is easy to see now that in this CA every particle moves uninterrupted in its direction, and there are no interactions between particles. Note that each "super cell" can contain up to four particles, all moving to different directions.

Next, let us introduce particle interaction in the case when two particles collide head-on. The new permutation π is the following:



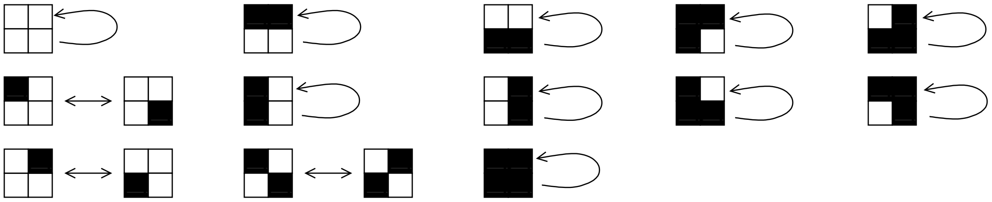
The only change to the previous rule is in a block with two diagonally aligned black and white cells: In such head-on collision both particles turn 90°. The resulting CA is the HPP lattice gas. Our sample configuration of three particles evolves now as follows:



HPP is a simple example, intended to simulate gas or fluid dynamics. The particles represent molecules in the simulated substance, and the local rule models the interaction of colliding molecules. In simulations very large numbers of particles are used, which produces a "realistic" behavior. HPP is reversible as is the physical system it attempts to simulate. HPP also has *conservation laws* that are familiar from physics: (1) The total number of particles remains invariant since the left- and the right-hand-sides of the update rules have the same particle counts. This also says that the total energy of the system is conserved, since each particle has the same kinetic energy. (2) The total momentum of the system is preserved. By momentum we mean the sum of the velocity vectors of the particles. Again, this is immediately seen from the update rules when each black cell is attached the velocity of the particle. These (number of particles, total momentum of the particles) are examples of *conserved quantities*, which will be studied in more details in the next section. Note that conservation laws are important in physics since they provide means to write equations. c It turns out that HPP is not a sufficiently realistic model of fluid or gas behavior. Because of the underlying square lattice, HPP is *anisotropic*: Different directions of the space have different characteristics. For example waves move differently in directions that match with the lattice than in other directions. This problem is remedied by changing square lattice into a hexagonal lattice. Now there are six possible directions for the particles. So-called FHP lattice gas is an example of such more advanced lattice gas.

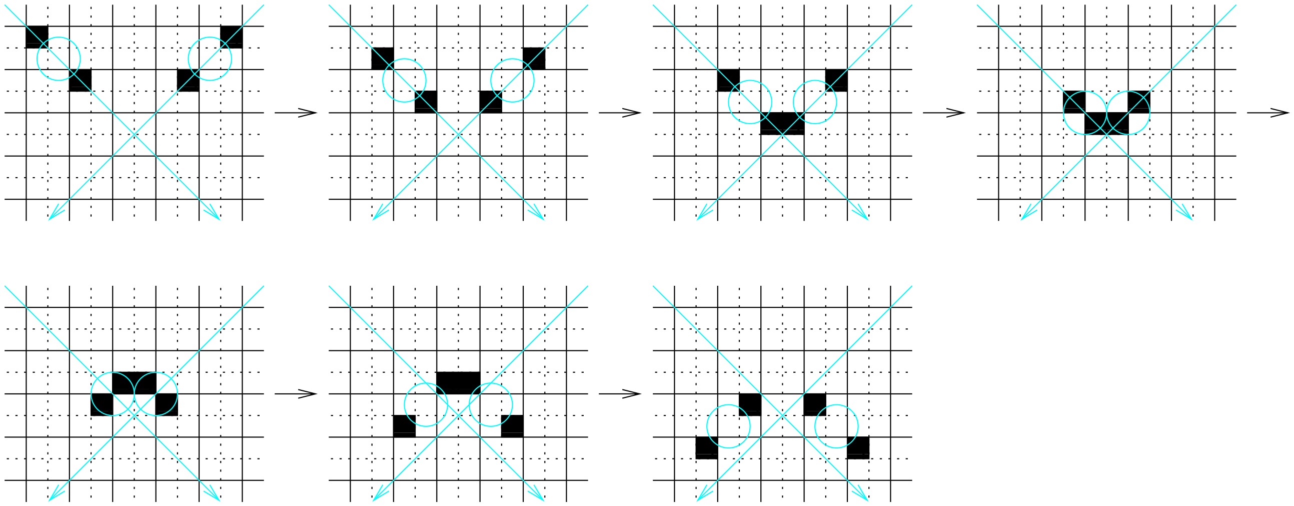
A problem with HPP is also that there are extra conservation laws not present in the physical system. For example, the total momentum of particles along each individual diagonal is preserved.

Another interesting example that uses the Margolus neighborhood is a CA that simulates the billiard ball model of computation (BBM). In this case the local update rule is based on the bijection



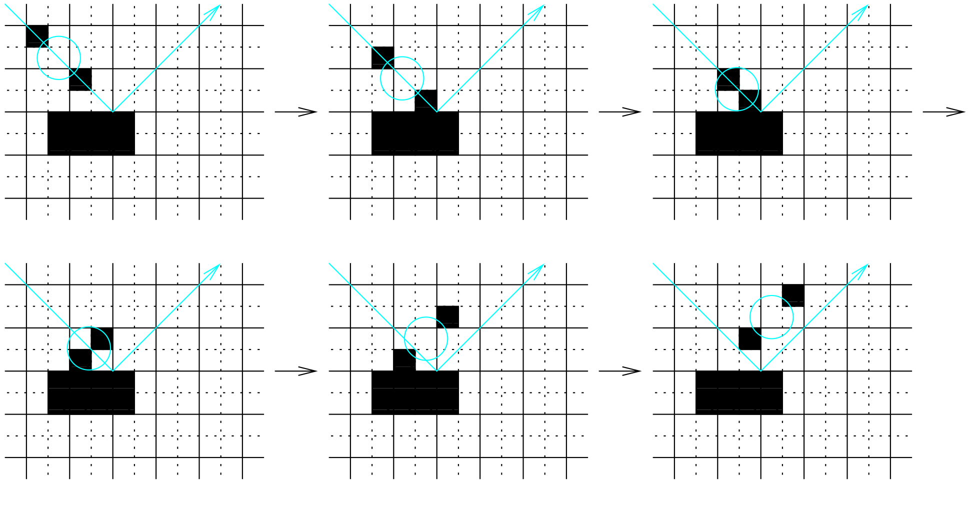
Note that again the numbers of black and white cells are conserved. The CA gets its name from the fact that it can simulate collisions between balls of positive radius. Also bouncing of such balls from static walls can be simulated. A ball will be represented by two particles moving to the same direction along the same trajectory. The distance of the particles determines the radius of the ball. In the following examples we use smallest possible balls, i.e. the second particle follows the first one time step behind.

The following figure illustrates what happens when two balls collide.



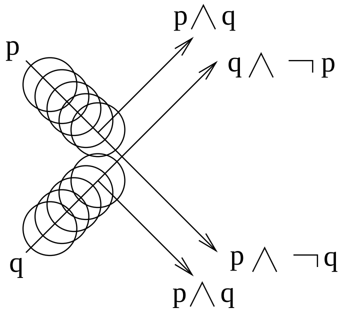
The light arrows indicate the trajectories of the two balls if no collision would occur. Notice that the collision translates the trajectories exactly as a collision of two billiard balls would alter their trajectories. Observe also that the collision is not perfectly simulated in the sense that the balls stop for a moment during the collision, so the timing is not identical to the physical collision of two balls.

A mirror is a stable pattern from which balls bounce. A properly placed two-by-four block of black cells does the trick:

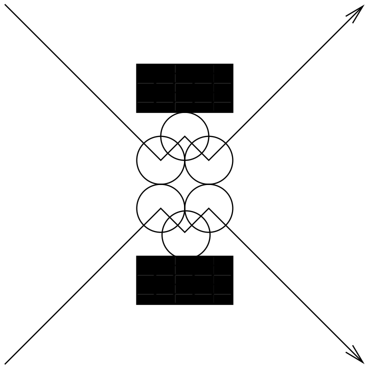


Now the light arrow indicates the path that a moving point would take. Note also a time delay in this bounce.

An interesting fact about collisions of positive radius balls is that one can implement logical gates, and then combine these gates to implement boolean circuits. Let us represent wires as potential trajectories of balls, and intersection points of the trajectories implement logical gates. Logical state "1" is represented by the presence of a ball while state "0" is indicated by the absence of a ball. Then, for example, two intersecting trajectories have the following effect:

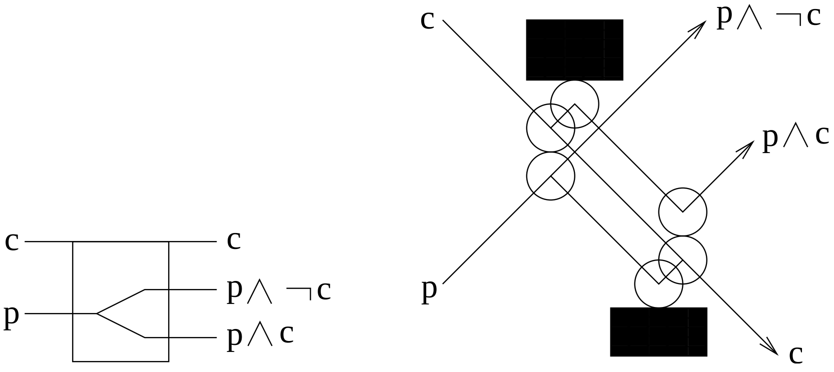


Two wires can cross if two walls are placed appropriately:



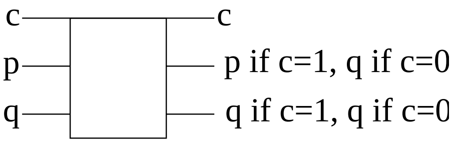
Wires can be turned using walls, and delays can be created by increasing the length of the wire by additional turns.

Let us build some additional logic gates. The first gate is a switch gate that performs conditional routing. The following figure shows our notation for the gate and its implementation using balls and mirrors:

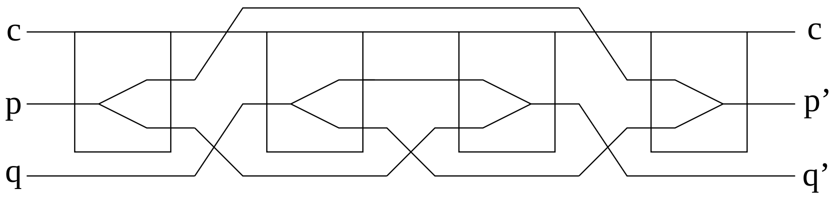


Note that we can use the switch gate in the opposite direction to select between two inputs, under the condition that the non-selected input is 0.

As a final example, consider the *Fredkin gate*. It is a controlled switch gate with three inputs and corresponding outputs. If the first input is "0" then the the other two signals are swapped, otherwise all outputs are the same as the corresponding input:



It can be implemented using four switch gates, two of which are used in the opposite direction:



Note that in this construction trajectories cross so we need mirrors as discussed above. Also, delay units may be needed to adjust the timing of the balls in the switch gates. Delays may also be used to make the timing of the output to be independent of input values.

The Margolus neighborhood can be naturally generalized in several ways. It can be used in other dimensions than $d = 2$. For example, in one-dimensional space we would partition the line into segments of length two, apply a bijective function $\pi : S^2 \longrightarrow S^2$ in each segment, and repeat the operation using a partitioning that is translated by one cell.

One modification is to use different block sizes. One can also use different bijections π_1 and π_2 on different partitionings. Finally, one can also increase from two the number of partitionings and rounds in one iteration of the rule. In all cases it is easy to guarantee reversibility by making sure that the block transforms are bijective, and it is equally easy to force conservation laws in the system.

4.3 Conserved quantities

As discussed earlier, additive conserved quantities play an important role in physics. Let G be a d-dimensional CA function over state set S. An *additive quantity* is any function

$$\mu : S \longrightarrow \mathbb{R}$$

that assigns a real value to each state. We extend μ to configurations by summing up the values assigned to states of cells, over all cells. However, a problem arises from the fact that the sum is infinite. We have two approaches to overcome the problem: Either we only consider q-finite configurations where q is the quiescent state satisfying $\mu(q) = 0$, or we consider totally periodic configurations and make the sum over one period only.

First approach: Let q be a quiescent state of G and assume μ satisfies $\mu(q) = 0$. Then for any finite configuration c we define

$$\hat{\mu}_F(c) = \sum_{\vec{n} \in \mathbb{Z}^d} \mu(c(\vec{n})).$$

Note that the sum has only a finite number of non-zero values. We say that μ is conserved on finite configurations by G if and only if for all finite configurations c holds

$$\hat{\mu}_F(G(c)) = \hat{\mu}_F(c).$$

Second approach: For every totally periodic configuration c we define

$$\hat{\mu}_P(c) = \frac{1}{k^d} \sum_{\vec{n} \in C} \mu(c(\vec{n}))$$

where k is such that c is σ_i^k invariant for all $i = 1, 2, \ldots d$, and C is a hypercube of size k^d. Note that for any such k and C we obtain the same value of $\hat{\mu}_P(c)$. It is the average value in c over all cells. We say that μ is conserved on periodic configurations by G if and only if for all totally periodic configurations c holds

$$\hat{\mu}_P(G(c)) = \hat{\mu}_P(c).$$

Let us first show that the two ways to define conserved quantities are equivalent:

Proposition 54 *Let G be a CA with quiescent state q and let μ be an additive quantity satisfying $\mu(q) = 0$. Then μ is conserved on periodic configurations if and only if μ is conserved on finite configurations.*

Proof. "$\Longrightarrow$" Suppose μ is conserved on periodic configurations and let c be an arbitrary finite configuration. Consider a hypercube C that contains all non-quiescent cells as well as all cells that have a non-quiescent neighbor in c. Extract the pattern with domain C from configuration c and let p be the totally periodic configuration where this pattern repeats. Then

$$\hat{\mu}_P(p) = \frac{1}{|C|} \hat{\mu}_F(c)$$

and

$$\hat{\mu}_P(G(p)) = \frac{1}{|C|} \hat{\mu}_F(G(c)).$$

But then

$$\hat{\mu}_F(G(c)) = |C| \hat{\mu}_P(G(p)) = |C| \hat{\mu}_P(p) = \hat{\mu}_F(c),$$

so μ is conserved on finite configurations.

"$\Longleftarrow$" Suppose μ is conserved on finite configurations and let p be an arbitrary totally periodic configuration. Let C be a hypercube of size k^d where k is such that p is invariant under translations σ_i^k for all $i = 1, 2, \ldots, d$. We may also assume that k is larger than the radius of the CA that defines G. Let j be a large positive integer, and construct a finite configuration c that consists of a size $(jk)^d$ hypercube extracted from p, with copies of the quiescent state q outside the hypercube. Then

$$\hat{\mu}_F(c) = (jk)^d \hat{\mu}_P(p).$$

Configurations $G(c)$ and $G(p)$ agree inside a size $(jk - 2k)^d$ hypercube, and in $G(c)$ all cells outside the co-centric size $(jk + 2k)^d$ hypercube are quiescent. Let

$$m = \max\{|\mu(s)| \mid s \in S\}$$

be the maximum absolute value assigned to any state. We have

$$\left|\hat{\mu}_F(G(c)) - (jk)^d \hat{\mu}_P(G(p))\right| \leq 2m\left[(jk+2k)^d - (jk-2k)^d\right] \leq 2m\left[4dk(jk+2k)^{d-1}\right].$$

Because μ is conserved on finite configurations,

$$\hat{\mu}_F(G(c)) = \hat{\mu}_F(c) = (jk)^d \hat{\mu}_P(p),$$

so we have

$$|\hat{\mu}_P(p) - \hat{\mu}_P(G(p))| \leq \frac{8dkm(jk+2k)^{d-1}}{(jk)^d} \longrightarrow 0$$

when $j \longrightarrow \infty$. This means that

$$\hat{\mu}_P(p) = \hat{\mu}_P(G(p)).$$

□

Note that the restriction that $\mu(q) = 0$ is not important: if μ is conserved on periodic configurations so is μ' where $\mu'(s) = \mu(s) + c$ for some constant c. So if $\mu(q) \neq 0$ we can consider $\mu'(s) = \mu(s) - \mu(q)$ instead. It satisfies the constraint $\mu'(q) = 0$ and it is conserved on periodic configurations if and only if μ is.

Since the two concepts of conservation are equivalent we concentrate on finite configurations in the following. From now on we denote briefly $\hat{\mu}$ for $\hat{\mu}_F$, and say that the quantity is conserved if it is conserved on finite configurations.

The following proposition provides an efficient characterization for conserved quantities.

Proposition 55 *Quantity μ is conserved if and only if for any two finite configurations c_1 and c_2 that differ in a single cell holds*

$$\hat{\mu}(c_1) - \hat{\mu}(c_2) = \hat{\mu}(G(c_1)) - \hat{\mu}(G(c_2)). \tag{8}$$

Proof. If μ is conserved then $\hat{\mu}(G(c_1)) = \hat{\mu}(c_1)$ and $\hat{\mu}(G(c_2)) = \hat{\mu}(c_2)$ so the given equation holds. Conversely, suppose the given equation holds, and let c be an arbitrary finite configuration. Then there exists a finite sequence of configurations $c_1, c_2, \ldots, c_n$ where c_1 is the quiescent configuration, $c_n = c$, and configurations c_i and c_{i+1} differ in a single cell for all $i = 1, 2, \ldots, n-1$. By the hypothesis we have

$$\hat{\mu}(c_i) - \hat{\mu}(c_{i+1}) = \hat{\mu}(G(c_i)) - \hat{\mu}(G(c_{i+1}))$$

for all $i = 1, 2, \ldots, n-1$. Adding all these equations together gives

$$\hat{\mu}(c_1) - \hat{\mu}(c_n) = \hat{\mu}(G(c_1)) - \hat{\mu}(G(c_n))$$

Since $G(c_1) = c_1$ and $c_n = c$, we obtain

$$\hat{\mu}(G(c)) = \hat{\mu}(c).$$

□

Let us look at equations (8) in detail. Since $\hat{\mu}$ is translation invariant, it is sufficient to consider only configurations c_1 and c_2 that differ in cell $\vec{0}$ and agree at all other cells. In the difference on the left-hand-side of (8) most values cancel out, so in fact

$$\hat{\mu}(c_1) - \hat{\mu}(c_2) = \mu(c_1(\vec{0})) - \mu(c_2(\vec{0})).$$

Analogously, since $G(c_1)$ and $G(c_2)$ agree on all cells but those that have $\vec{0}$ in their neighborhood, the right-hand-side of (8) becomes

$$\hat{\mu}(G(c_1)) - \hat{\mu}(G(c_2)) = \sum_{\vec{n} \in A} \mu(G(c_1)(\vec{n})) - \mu(G(c_2)(\vec{n}))$$

where

$$A = \{-\vec{n}_i \mid i = 1, 2, \ldots, m\}$$

is the set of all cells that have $\vec{0}$ as a neighbor. From this we already see the interesting fact that there are only a finite number of different equations (8).

Let c_1 and c_2 be two finite configurations that only differ in cell $\vec{0}$, let c_1' and c_2' be another such pair, and suppose that c_1 and c_1' agree with each other inside region

$$B = \{\vec{n}_j - \vec{n}_i \mid i, j = 1, 2, \ldots, m\},$$

and suppose that c_2 and c_2' also agree inside B. Notice that region B contains all cells that are neighbors of elements of A, so $G(c_1)$ and $G(c_1')$ agree in A, and also $G(c_2)$ and $G(c_2')$ agree in A. Then the equation (8) we obtain using c_1 and c_2 is the same as the equation that we obtain if we use c_1' and c_2' instead. This means that it is enough to form the equations (8) for different patterns with domain B. In other words, for all pairs $p_1 = (B, g_1)$ and $p_2 = (B, g_2)$

of patterns with domain B, where $g_1(\vec{n}) = g_2(\vec{n})$ for all $\vec{n} \neq \vec{0}$ and $g_1(\vec{0}) \neq g_2(\vec{0})$, we calculate the successors $(A, h_1) = G(p_1)$ and $(A, h_2) = G(p_2)$, and form the corresponding equation

$$\mu(g_1(\vec{0})) - \mu(g_2(\vec{0})) = \sum_{\vec{n} \in A} \mu(h_1(\vec{n})) - \mu(h_2(\vec{n})).$$

Quantity μ is conserved by G if and only if it satisfies all these equations. In addition, requirement $\mu(q) = 0$ may be added where q is the quiescent state, although this is not necessary since $\mu(q) = 0$ only scales the quantity by an additive constant.

If the CA has no quiescent state q we may pick an arbitrary ground state q and consider q-finite configurations. The considerations above work unaltered, but we have to add the equation

$$\mu(q) = \mu(f(q, q, \ldots, q))$$

stating that the quantity is conserved on the configuration where all cells are in state q.

In fact, we can reduce the number of equations further by a more careful inspection of the proof of Proposition 55. For simplicity, let us consider one-dimensional CA only (although a similar analysis works for higher dimensional cases as well). The quiescent configuration can be transformed into any desired finite configuration by changing states of cells one-by-one in the left-to-right order. This means that configurations c_1 and c_2 in the equation (8) satisfy the additional constraint that all cells to the right of the position where c_1 and c_2 differ are quiescent. This means that it is enough to form equations using patterns $p_1 = (B, g_1)$ and $p_2 = (B, g_2)$ that satisfy the following constraints (in the one-dimensional case):

- $g_1(n) = g_2(n) = q$ for all $n > 0$,
- $g_1(0) = q$ and $g_2(0) \neq q$, and
- $g_1(n) = g_2(n)$ for all $n < 0$.

Example 26. As an example, let us find conservation laws of the traffic CA, that is, the elementary CA with Wolfram number 226. The local rule replaces pattern 01 by pattern 10:

$$\begin{array}{lllllllllllll} 000 & \mapsto & 0 & \quad 001 & \mapsto & 1 & \quad 010 & \mapsto & 0 & \quad 011 & \mapsto & 0 \\ 100 & \mapsto & 0 & \quad 101 & \mapsto & 1 & \quad 110 & \mapsto & 1 & \quad 111 & \mapsto & 1 \end{array}$$

The neighborhood vector is $N = (-1, 0, 1)$, so $A = \{-1, 0, 1\}$ and $B = \{-2, -1, 0, 1, 2\}$. This means that changing the state of cell 0 may affect the states of cells -1,0 and 1, and how these are affected depends on the old states of cells in the set B. It is hence sufficient to form the equations based on patterns with domain B. As discussed above, it is in fact sufficient to form equations (8) based on pairs of patterns $ab000$ and $ab100$ for all $a, b \in \{0, 1\}$.

For $a = 0, b = 0$ we obtain from

$$\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & \end{array} \qquad\qquad \begin{array}{ccccc} 0 & 0 & 1 & 0 & 0 \\ & 1 & 0 & 0 & \end{array}$$

the equation

$$\mu(0) - \mu(1) = 3\mu(0) - [\mu(1) + 2\mu(0)].$$

This yields the trivial relation $0 = 0$. For $a = 0, b = 1$ we get from

$$\begin{array}{ccccc} 0 & 1 & 0 & 0 & 0 \\ & 0 & 0 & 0 & \end{array} \qquad\qquad \begin{array}{ccccc} 0 & 1 & 1 & 0 & 0 \\ & 0 & 1 & 0 & \end{array}$$

also the trivial equation $0 = 0$. For $a = 1, b = 0$ we have

$$\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & \end{array} \qquad\qquad \begin{array}{ccccc} 1 & 0 & 1 & 0 & 0 \\ & 1 & 0 & 0 & \end{array}$$

which is again trivial, and finally, for $a = b = 1$ we have

$$\begin{array}{ccccc} 1 & 1 & 0 & 0 & 0 \\ & 1 & 0 & 0 & \end{array} \qquad\qquad \begin{array}{ccccc} 1 & 1 & 1 & 0 & 0 \\ & 1 & 1 & 0 & \end{array}$$

which also yields the trivial equation. We conclude that in traffic CA all functions μ are conserved. In particular, number of cells in state 1 is conserved. □

Example 27. Consider the CA discussed in Example 11. It uses the radius-$\frac{1}{2}$ neighborhood $N = (0, 1)$, so $A = \{-1, 0\}$ and $B = \{-1, 0, 1\}$. The state set is $S = \{0, 1, 2\}$ and the local rule f is

$$f(a, b) = \begin{cases} 2, & \text{if } a = 2, \\ 0, & \text{if } a \neq 2 \text{ and } a + b \text{ is even, and} \\ 1, & \text{if } a \neq 2 \text{ and } a + b \text{ is odd.} \end{cases}$$

Now it is enough to consider pairs of patterns $a00$ and $ab0$ for $a, b \in S$, $b \neq 0$.

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & \end{array} \qquad\qquad \left.\begin{array}{ccc} 0 & 1 & 0 \\ 1 & 1 & \end{array}\right\} \Longrightarrow \mu(0) - \mu(1) = 2\mu(0) - 2\mu(1) \Longrightarrow \mu(0) = \mu(1)$$

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & \end{array} \qquad\qquad \left.\begin{array}{ccc} 0 & 2 & 0 \\ 0 & 2 & \end{array}\right\} \Longrightarrow \mu(0) - \mu(2) = 2\mu(0) - \mu(0) - \mu(2) \Longrightarrow 0 = 0$$

$$\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 0 & \end{array} \qquad\qquad \left.\begin{array}{ccc} 1 & 1 & 0 \\ 0 & 1 & \end{array}\right\} \Longrightarrow \mu(0) = \mu(1)$$

$$\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 0 & \end{array} \qquad\qquad \left.\begin{array}{ccc} 1 & 2 & 0 \\ 1 & 2 & \end{array}\right\} \Longrightarrow 0 = 0$$

$$\begin{array}{ccc} 2 & 0 & 0 \\ 2 & 0 & \end{array} \qquad\qquad \left.\begin{array}{ccc} 2 & 1 & 0 \\ 2 & 1 & \end{array}\right\} \Longrightarrow 0 = 0$$

$$\begin{array}{ccc} 2 & 0 & 0 \\ 2 & 0 & \end{array} \qquad\qquad \left.\begin{array}{ccc} 2 & 2 & 0 \\ 2 & 2 & \end{array}\right\} \Longrightarrow 0 = 0$$

We conclude that any μ satisfying $\mu(0) = \mu(1)$ is conserved. In particular, the number of 2's is conserved.

□

As a final note, we observe that for any fixed CA, the additive quantities that it conserves form a linear space, when sum and scalar product are defined in the natural way, point wise:

$$\begin{array}{rcll} (\mu+\mu')(a) & = & \mu(a)+\mu'(a) & \text{for all } \mu,\mu' \in \mathbb{R}^S \text{ and } a \in S, \\ (r\mu)(a) & = & r\mu(a) & \text{for all } \mu \in \mathbb{R}^S,\ r \in \mathbb{R} \text{ and } a \in S. \end{array}$$

If G conserves μ and μ' then for any q-finite configuration c and every $r \in \mathbb{R}$ holds

$$(\mu+\mu')(c) = \mu(c)+\mu'(c) = \mu(G(c)) + \mu'(G(c)) = (\mu+\mu')(G(c))$$

and

$$(r\mu)(c) = r\mu(c) = r(\mu(G(c)) = (r\mu)(G(c)),$$

which proves the linearity of the conserved quantities of G. Note also that equations (8) are linear, so we have an algorithm to build a system of linear equations whose solutions are exactly the conserved quantities of G.

5 Cellular automata dynamical systems

An especially fruitful view to cellular automata dynamics is to endow the configuration space $S^{\mathbb{Z}^d}$ with a metric under which CA functions G are continuous. This makes the tools and results of topological dynamics available to the analysis of cellular automata. The space $S^{\mathbb{Z}^d}$ under this metric is compact and complete. Convergence of a sequence $c_1, c_2, \ldots$ of elements under this metric is exactly equivalent to the convergence introduced in Section 1.8. The compactness principle (Proposition 4) of that section then simply states the compactness of the metric space.

First, for any $\vec{x} \in \mathbb{Z}^d$ we denote

$$\|(x_1, x_2 \ldots, x_d)\| = \max\{|x_1|, |x_2|, \ldots, |x_d|\}.$$

See Figure 42 for an illustration in the two dimensional case. Then, we define the distance $d(e,c)$ between configurations $e, c \in S^{\mathbb{Z}^d}$ as follows:

$$d(e,c) = \begin{cases} 0, & \text{if } e = c, \\ 2^{-\min\{\|\vec{x}\| \ \mid\ c(\vec{x}) \neq e(\vec{x})\}}, & \text{if } e \neq c. \end{cases}$$

In other words, two configurations that differ in a cell that is close to $\vec{0}$ are far away from each other under this metric, while configurations that agree with each other on a large area around the origin are close to each other. Under this metric, two configurations have distance $< 2^{-r}$ if and only if they agree with each other at all positions (x,y) where $\|\vec{x}\| \leq r$.

Note that other vector norms $\|\cdot\|$ could be used as well, and any other decreasing function whose limit at ∞ is 0 could be used instead of $x \mapsto 2^{-x}$. A different metric, but the same topology would result.

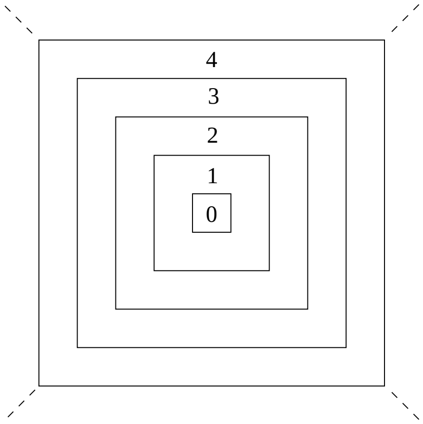


Figure 42: The value of $\|\vec{x}\|$ in the two-dimensional case.

Lemma 56 *Function $d : S^{\mathbb{Z}^d} \times S^{\mathbb{Z}^d} \longrightarrow \mathbb{R}$ is a metric.*

Proof. We have to check the three defining properties of metric:

(a) $d(c, e) \geq 0$, and $d(c, e) = 0$ if and only if $c = e$,

(b) $d(c, e) = d(e, c)$, and

(c) $d(c, e) \leq d(c, c') + d(c', e)$.

The first two conditions (a) and (b) are immediate. The third condition (c), called the triangle inequality, follows from the fact that for every $\vec{x} \in \mathbb{Z}^d$, if $c(\vec{x}) \neq e(\vec{x})$ then either $c(\vec{x}) \neq c'(\vec{x})$ or $c'(\vec{x}) \neq e(\vec{x})$, or both. This means that either $d(c, c') \geq d(c, e)$ or $d(c', e) \geq d(c, e)$, so even the strong form

$$d(c, e) \leq \max\{d(c, c'), d(c', e)\}$$

of the triangle inequality holds. □

From now on we consider $S^{\mathbb{Z}^d}$ as a metric topological space under this metric. The following subsection contains a brief review of some basic facts about metric spaces.

5.1 Review of topology and metric spaces

Let X be a set. A family $\mathcal{T}$ of subsets of X is called a *topology* if it satisfies the following three conditions:

(i) $\emptyset \in \mathcal{T}$ and $X \in \mathcal{T}$,

(ii) the union of the sets in any subfamily of $\mathcal{T}$ is in $\mathcal{T}$,

(iii) the intersection of $\underline{\text{finitely}}$ many elements of $\mathcal{T}$ is always in $\mathcal{T}$.

Elements of $\mathcal{T}$ are called *open* sets, and their complements (with respect to X) are *closed* sets. A set that is both open and closed is called *clopen*.

Example 28. For any X, let $\mathcal{T}$ contain all subsets of X. Then $\mathcal{T}$ is a topology, the *discrete* topology of X. Also $\{X, \emptyset\}$ is a topology, the *trivial* topology of X. □

Example 29. Let us call $S \subseteq \mathbb{R}$ open if for every $x \in S$ there is a positive real $\varepsilon > 0$ such that $|y - x| < \varepsilon \implies y \in S$. These open sets form a topology of $X = \mathbb{R}$. It is called the usual topology of $\mathbb{R}$. For example, all open intervals (a, b) for $a < b$ are open sets. Closed intervals $[a, b]$ are not open but they are closed. Set $\mathbb{Q}$ of rational numbers is not open or closed. The only clopen sets are $\emptyset$ and $\mathbb{R}$. □

Generalizing the previous example, let X be a set and let $d : X \times X \longrightarrow \mathbb{R}$ be a metric. For every $\varepsilon > 0$ and $x \in X$ we denote

$$B_\varepsilon(x) = \{y \in X \mid d(x, y) < \varepsilon\}$$

and call $B_\varepsilon(x)$ the (open) ε-ball with center x. Let us call $U \subseteq X$ open if

$$\forall x \in U \ : \ \exists \varepsilon > 0 \ : \ B_\varepsilon(x) \subseteq U.$$

These open sets form a topology of X, the *metric topology* induced by d.

Example 30. The discrete topology is induced by the discrete metric

$$d(x, y) = \begin{cases} 0, & \text{if } x = y, \\ 1, & \text{if } x \neq y. \end{cases}$$

In contrast, if $|X| \geq 2$ then the trivial topology $\{X, \emptyset\}$ is not metric. □

Let $A \subseteq X$. Point $x \in X$ is an *accumulation point* of A if every open set U that contains x also contains some element $y \neq x$ of A. The following simple properties hold for closed sets:

Proposition 57 *A subset $A \subseteq X$ is closed if and only if its accumulation points belong to A. Closed sets satisfy the following properties (that are dual statements of the defining properties of open sets):*

(i) The empty set $\emptyset$ is closed, and X is closed,

(ii) the intersection of any number of closed sets is closed, and

(iii) the union of a finite number of closed sets is closed.

□

Let $A \subseteq X$. The *closure* of A is the intersection of all closed sets that contain A. It is then the smallest closed set that contains A. We denote the closure of A by $\overline{A}$. Notice that A itself is closed if and only if $\overline{A} = A$. Notice also that the closure of A is the union of A and its set of accumulation points.

Set A is called *dense* if $\overline{A} = X$.

Example 31. Consider the usual topology of $\mathbb{R}$. All real numbers are accumulation points of the set $\mathbb{Q}$ of rational numbers. This means that the closure of $\mathbb{Q}$ is $\mathbb{R}$, so $\mathbb{Q}$ is dense in $\mathbb{R}$. Accumulation points of the open interval $(0, 1)$ are the elements of the closed interval $[0, 1]$, while the set $\mathbb{Z}$ of integers has no accumulation points. □

Let $A \subseteq X$. Point $x \in A$ is an *interior point* of A if there is an open set U such that $x \in U$ and $U \subseteq A$. The set A° of all interior points of A is the *interior* of A. It is easily seen to be the union of all open subsets of A, or equivalently, the largest open subset of A. Then set A is open if and only if its interior is A itself.

The *exterior* of set $A \subseteq X$ is the interior of the complement of A, and the *boundary* of A consists of all points that are not in the interior or the exterior of A. Note that the interior, exterior and boundary of A is a partitioning of X. A set $A \subseteq X$ is called a *neighborhood* of $x \in X$ if x is an interior point of A, that is, if there is an open set U such that $x \in U \subseteq A$.

Example 32. In the usual topology of $\mathbb{R}$, the interior, exterior and the boundary of an open interval (a, b) are (a, b), $(-\infty, a) \cup (b, \infty)$ and $\{a, b\}$, respectively. The closed interval $[a, b]$ has these same interior, exterior and boundary. Set $\mathbb{Q}$ has empty interior and exterior. All real numbers are in its boundary. □

A topology is called *Hausdorff* if for every $x \neq y$ there are open U_x and U_y such that $x \in U_x$, $y \in U_y$ and $U_x \cap U_y = \emptyset$. In other words, any two distinct points have non-intersecting neighborhoods.

Example 33. Every metric topology is Hausdorff. Indeed, if $x \neq y$ then $d(x, y) > 0$. If we choose $\varepsilon = \frac{1}{2}d(x, y)$ then $B_\varepsilon(x)$ and $B_\varepsilon(y)$ are non-intersecting neighborhoods of x and y □

A sequence $x_1, x_2, \ldots$ of points of X *converges* to point $x \in X$ if for every open $U \subseteq X$ that contains x there is positive integer n such that $x_i \in U$ for all $i \geq n$. If the topology is metric this is equivalent to saying that for every $\varepsilon > 0$ there is n such that $d(x_i, x) < \varepsilon$ for all $i \geq n$.

Note that generally a converging sequence may converge to several different points, but if the topology is Hausdorff (e.g. metric) the limit is unique.

Proposition 58 *In Hausdorff topology every converging sequence converges to a unique point.* □

Proof. Suppose $x_1, x_2, \ldots$ converges to x and y where $x \neq y$. Since X is Hausdorff, there are open sets U and V such that $x \in U$, $y \in V$ and $U \cap V = \emptyset$. By the definition of convergence, $x_i \in U$ and $x_i \in V$ for all sufficiently large i, a contradiction. □

Note: the proposition does not hold in all topological spaces. For example, in the trivial topology $\mathcal{T} = \{\emptyset, X\}$ every sequence converges to every point.

In Hausdorff topology we denote by $\lim_{i\to\infty} x_i$ the unique point into which the sequence $x_1, x_2, \ldots$ converges, if it exists. This point is the limit of the sequence.

The following proposition states that if the topology is metric then the closure $\overline{A}$ of any set A consists exactly of the limits of converging sequences of elements of A:

Proposition 59 *Let X be a metric space and $A \subseteq X$. Then $x \in \overline{A}$ if and only if $x = \lim_{i\to\infty} a_i$ for some converging sequence $a_1, a_2, \ldots$ where all $a_i \in A$.*

Proof. "$\Longleftarrow$": Let $a_1, a_2, \ldots$ be a converging sequence where all $a_i \in A$ and let $x = \lim_{i\to\infty} a_i$. Let U be an arbitrary open set that contains x. By the definition of convergence there are some $a_i \in U$, so $U \cap A \neq \emptyset$. This means that $x \in \overline{A}$. (This direction of the proof holds for any topological space.)

"$\Longrightarrow$": Conversely, suppose $x \in \overline{A}$. For every positive integer i, let a_i be an element of $A \cap B_{\frac{1}{i}}(x)$. Then $d(x, a_i) < \frac{1}{i}$, so $x = \lim_{i\longrightarrow\infty} a_i$. □

Corollary 60 *In metric space X, set A is closed if and only if it contains the limit of every converging sequence of its elements.* □

A family $\mathcal{B}$ of open sets is called a *base* of the topology iff every open set is the union of some members of $\mathcal{B}$. Equivalently: $\mathcal{B} \subseteq \mathcal{T}$ is a base if for every open set U and $x \in U$ there exists some $B \in \mathcal{B}$ with the property that $x \in B \subseteq U$.

Example 34. The open intervals (a, b) with $a < b$ form a base of the usual topology of $\mathbb{R}$. More generally, in any metric topology the open balls $B_\varepsilon(x)$ over all $\varepsilon > 0$ and $x \in X$ form a base. □

If $\mathcal{B}$ is a base of a topology then this topology is uniquely determined by $\mathcal{B}$: open sets are exactly the unions of members of $\mathcal{B}$. The following proposition gives a necessary and sufficient condition on when a family $\mathcal{B}$ is a base of some topology:

Proposition 61 *$\mathcal{B}$ is a base of a topology if and only if it satisfies the following two conditions:*

(i) Every $x \in X$ belongs to some $B \in \mathcal{B}$, and

(ii) For every $A, B \in \mathcal{B}$ and every $x \in A \cap B$ there is $C \in \mathcal{B}$ such that $x \in C \subseteq A \cap B$.

Proof. Suppose first that $\mathcal{B}$ is a base of a topology. Condition (i) follows from the fact that X is open. Let us prove (ii). Let $A, B \in \mathcal{B}$ be arbitrary, and let $x \in A \cap B$. Because A and B are both open, so is $A \cap B$. Hence there is a base set $C \in \mathcal{B}$ that satisfies $x \in C \subseteq A \cap B$.

Conversely, suppose (i) and (ii) hold, and let $\mathcal{T}$ contain all unions of all subfamilies of $\mathcal{B}$. The empty union is $\emptyset$ so $\emptyset \in \mathcal{T}$. It follows from (i) that the union of all elements in $\mathcal{B}$ is X, so $X \in \mathcal{T}$.

Let U_i be arbitrary elements of $\mathcal{T}$. Then each U_i is a union of elements of $\mathcal{B}$, so their union is also a union of elements of $\mathcal{B}$. Hence the union of U_i's is in $\mathcal{T}$.

Consider then arbitrary $U, V \in \mathcal{T}$, and let $x \in U \cap V$. Because $x \in U$, and U is a union of members of $\mathcal{B}$, there is some $A \in \mathcal{B}$ such that $x \in A \subseteq U$. Analogously there is some $B \in \mathcal{B}$ such that $x \in B \subseteq V$. By property (ii) there is $C \in \mathcal{B}$ such that $x \in C \subseteq A \cap B \subseteq U \cap V$. Such C exists for every x, so we see that $U \cap V \in \mathcal{T}$. This means that the intersection of finitely many elements of $\mathcal{T}$ is in $\mathcal{T}$. □

Next we define compactness. Let $A \subseteq X$ where X is a topological space. A family of open sets U_i is called an *open cover* of A if every element of A belongs to some U_i. A subfamily of an open cover of A is called a *subcover* if it is also a cover of A.

Set $A \subseteq X$ is called *compact* if every open cover of A has a finite subcover of A. The topology is called compact if the whole space X is compact. In other words, a topology is compact if every family of open sets whose union is X has a finite subfamily whose union is X.

Example 35. In the usual topology of $\mathbb{R}$ the set

$$A = \{0\} \cup \{\frac{1}{n} \mid n \in \mathbb{Z}_+\}$$

is compact. Namely, an open set that contains 0 covers all but finitely many elements of A. So any open cover of A contains a finite subcover: Open set U that covers 0 together with a finite number of open sets that cover the finitely many elements of A that are outside of U.

On the other hand, set $B = \{\frac{1}{n} \mid n \in \mathbb{Z}_+\}$ is not compact. It has an open cover in which every open set covers exactly one element of B. Such cover has no finite subcover. □

The following proposition states the finite intersection property. It is dual to the open cover property we used as the definition, and in fact the finite intersection property could have been taken equally well as the definition of compactness. We state the property for the whole space X:

Proposition 62 *Topology of X is compact if and only if every family of closed sets whose intersection is empty has a finite subfamily whose intersection is empty.*

Proof. This follows directly from the definition of compactness and de Morgan's laws: A family of open sets is a cover of X if and only if the family of their complements have empty intersection. □

We typically apply the previous proposition in the following set-up:

Corollary 63 *Let $F_1 \supseteq F_2 \supseteq F_3 \supseteq \dots$ be an infinite chain of closed sets in a compact space X. If*

$$\bigcap_{i=1}^{\infty} F_i = \emptyset,$$

then $F_i = \emptyset$ for some i. □

The next proposition gives a characterization of compact subsets in metric spaces. The proposition gives a condition that looks very similar to Proposition 4 for configurations. In fact, we use the proposition later to show the compactness of the configuration space. The proposition is valid (and is stated) for arbitrary metric spaces, but we only prove it now for metric spaces that have a countable base. Our configuration space satisfies this restriction, so the proof is sufficient for our set-up. The proof for general metric spaces is not very difficult either.

Proposition 64 *Suppose X is a metric space. Set $A \subseteq X$ is compact if and only if every sequence $a_1, a_2, \dots$ of elements of A has a subsequence that converges to an element of A.*

Proof. "$\Longrightarrow$" Suppose A is compact, and let $a_1, a_2, \dots$ be arbitrary sequence where each $a_i \in A$.

Suppose first that there is some $a \in A$ such that for every $\varepsilon > 0$ the ball $B_\varepsilon(a)$ contains infinitely many different elements of the sequence $a_1, a_2, \dots$. Then the sequence has a subsequence that converges to a: There namely is a subsequence whose n'th element belongs to $B_{\frac{1}{n}}(a)$.

Suppose then that for every $a \in A$ there is some $\varepsilon_a > 0$ such that $B_{\varepsilon_a}(a)$ only contains finitely many different elements of the sequence $a_1, a_2, \dots$. Clearly the family of $B_{\varepsilon_a}(a)$ over all $a \in A$ is an open cover of A, so by compactness of A it has a finite subcover

$$U_i = B_{\varepsilon_{a_i}}(a_i) \text{ for } i = 1, 2, \dots m.$$

But each U_i only covers finitely many different elements of sequence $a_1, a_2, \dots$, while each element of the sequence is covered by some U_i. This means that the sequence has only finitely many different elements. Then some element $a \in A$ repeats infinitely many times in the sequence so the sequence has a constant subsequence $a, a, \dots$ which trivially converges to $a \in A$.

"$\Longleftarrow$" Suppose every sequence of elements of A has a converging subsequence whose limit is in A. Here we simplify the set-up by making the additional assumption that the topology has a countable base. Then it is enough to show that any countable open cover of A has a finite sub-cover. (Indeed, for an arbitrary open cover by U_i we can consider instead the countable cover that consists of all base sets B_j that are completely included in some U_i. If every countable cover has a finite subcover, then the original cover also has a finite subcover where we take for each selected B_j one U_i from the original cover that satisfies $B_j \subseteq U_i$.)

So consider a countable open cover $\{U_1, U_2, \ldots\}$ of A. If it has no finite subcover then for every i there is some $a_i \in A$ such that $a_i \notin U_j$ for all $j < i$. By the hypothesis, sequence $a_1, a_2, \ldots$ has a converging subsequence with limit $a \in A$. But $a \in U_j$ for some j, and then by the definition of convergence $a_i \in U_j$ for infinitely many indices i. In particular, there is $i > j$ such that $a_i \in U_j$, which contradicts the choice of a_i's. We conclude that a finite subcover must exist. □

Next two propositions show that in our forthcoming situation compact sets of the space are exactly the closed sets.

Proposition 65 *If X is a compact topological space then every closed $A \subseteq X$ is compact.*

Proof. Let $A \subseteq X$ be closed. Consider an open cover of A. Together with the complement of A it forms an open cover of X. By compactness of X this has a finite subcover of X, from which we obtain a finite subcover of A by removing the complement of A (if present). Hence A is compact. □

Proposition 66 *If X is Hausdorff then every compact $A \subseteq X$ is closed.*

Proof. Let $A \subseteq X$ be compact. Let $x \in X \setminus A$. By the Hausdorff property, for every $a \in A$ there are open sets U_a and V_a such that $a \in U_a$, $x \in V_a$ and $U_a \cap V_a = \emptyset$. Sets U_a form an open cover of A so by compactness of A there is a finite subcover $U_{a_1}, \ldots, U_{a_m}$ of A. But then the intersection

$$V_x = V_{a_1} \cap \ldots \cap V_{a_m}$$

of the corresponding sets V_{a_i} is an open set satisfying $x \in V_x$ and $V_x \cap A = \emptyset$. The union of sets V_x over all $x \in X \setminus A$ is the complement of A. Since the union is open, we see that A is closed. □

Let X be a metric space. A sequence $x_1, x_2, \ldots$ of points of X is a *Cauchy sequence* if for every $\varepsilon > 0$ there exists $n \geq 1$ such that $i, j > n \Longrightarrow d(x_i, x_j) < \varepsilon$. Metric space X is *complete* if every Cauchy sequence converges.

Example 36. Under the discrete metric, a sequence is a Cauchy sequence if and only if it is constant beyond some point, so the sequence converges. Discrete metric space is complete. The space $\mathbb{Q}$ of rational numbers under the usual metric is an example of a non-complete space. □

All compact metric spaces are complete:

Proposition 67 *A compact metric space is complete.*

Proof. Suppose X is a compact metric space, and let $x_1, x_2, \ldots$ be an arbitrary Cauchy sequence. By Proposition 64 it has a subsequence that converges to some $x \in X$. Let us prove the the whole sequence converges to x. For any $\varepsilon > 0$ there is some n such that $d(x_i, x_j) < \varepsilon/2$ for all $i, j > n$ (=Caychy property), and some $k > n$ such that $d(x_k, x) < \varepsilon/2$. This implies that

$$d(x_i, x) \leq d(x_i, x_k) + d(x_k, x) < \varepsilon/2 + \varepsilon/2 = \varepsilon$$

for all $i > n$. Hence the sequence $x_1, x_2, \ldots$ converges to x. □

A topological space is *separable* if it has a countable dense subset,and it is *second countable* if it has a countable base. Our space of interest is both separable and second countable. In fact, every compact metric space has these properties.

Proposition 68 *A compact metric space is separable.*

Proof. For every n the cover of X by the open balls $B_{1/n}(x)$ has a finite subcover. The centers of all the balls in these finite subcovers for $n = 1, 2, 3, \ldots$ form a countable set A. It is dense: For every $y \in X$ and $n \geq 1$ there is a ball $B_{1/n}(x)$ with center $x \in A$ that contains y. Then $x \in B_{1/n}(y)$. □

Proposition 69 *A separable metric space has a countable base.*

Proof. Let $\{x_1, x_2, \ldots\}$ be a dense countable subset of X. Then the open balls $B_{1/n}(x_i)$ over all positive integers i, n form a countable base. Indeed: For every open U and $x \in U$ there exists $\varepsilon > 0$ such that $B_\varepsilon(x) \subseteq U$. Choose an integer $n > 2/\varepsilon$. Some $x_i \in B_{1/n}(x)$. Because $1/n < \varepsilon/2$ we have

$$x \in B_{1/n}(x_i) \subseteq B_\varepsilon(x) \subseteq U.$$

□

A topological space X is called a *Baire space* if every countable intersection of dense open sets is dense. That is, if $U_1, U_2, \ldots$ are open sets such that for all i hold $\overline{U}_i = X$, then the set

$$A = \bigcap_{i=1}^{\infty} U_i$$

is dense. (In particular, A is non-empty.)

Proposition 70 *Every compact metric space is a Baire space.*

Proof. Let $U_1, U_2, \ldots$ be open dense sets, and let A be their intersection. Let U be an arbitrary non-empty open set. It is enough to prove that $U \cap A \neq \emptyset$. Let us define a sequence $V_0, V_1, V_2, \ldots$ of open sets as follows: $V_0 = U$, and for every $n \geq 1$, we choose as V_n

a non-empty, open set whose closure is a subset of $V_{n-1} \cap U_n$. Such V_n exists for the following reasons: Set $V_{n-1} \cap U_n$ is open, and non-empty by the denseness of U_n. This means that $B_\varepsilon(x) \subseteq V_{n-1} \cap U_n$ for some $\varepsilon > 0$ and $x \in X$. Then $B_{\varepsilon/2}(x)$ can be selected as V_n, because its closure is a subset of $B_\varepsilon(x)$.

Closures of V_n form a decreasing chain

$$\overline{V}_0 \supseteq \overline{V}_1 \supseteq \overline{V}_2 \supseteq \dots$$

of non-empty compact sets. By Corollary 63 their intersection is non-empty. The intersection is a subset of every U_n and also of U, so we conclude that $A \cap U \neq \emptyset$. □

Finally, a few words about continuous functions. Let X and Y be two topological spaces. A function $f : X \longrightarrow Y$ is *continuous* at point $x \in X$ if for every open $V \subseteq Y$ that contains $f(x)$ there exists an open $U \subseteq X$ such that $x \in U$ and $f(U) \subseteq V$.

If X and Y are metric spaces with metrics d and e, respectively, then continuity at x is equivalent to the following: For every $\varepsilon > 0$ there exists $\delta > 0$ such that $f(B_\delta(x)) \subseteq B_\varepsilon(f(x))$.

We call function $f : X \longrightarrow Y$ is *continuous* if it is continuous at every $x \in X$.

Example 37. If X has the discrete topology then every function $f : X \longrightarrow Y$ is continuous. Also, if Y has the trivial topology $\{\emptyset, Y\}$ then every $f : X \longrightarrow Y$ is continuous. In all topological spaces X and Y all constant functions $f : X \longrightarrow Y$ are continuous. If X has the trivial topology and Y has the discrete topology then the constant functions are the only continuous functions. □

Proposition 71 *Let $f : X \longrightarrow Y$ be a function between two topological spaces. The following conditions are equivalent:*

(i) Function $f : X \longrightarrow Y$ is continuous,

(ii) pre-image $f^{-1}(V)$ is open in X for every open $V \subseteq Y$,

(iii) pre-image $f^{-1}(C)$ is closed in X for every closed $C \subseteq Y$.

Proof. (i) $\Longrightarrow$ (ii): Suppose f is continuous and let $V \subseteq Y$ be open. Let $x \in f^{-1}(V)$ be arbitrary, so $f(x) \in V$. From continuity it follows that there is an open $U \subseteq X$ such that $f(U) \subseteq V$ and $x \in U$. This means that $x \in U \subseteq f^{-1}(V)$, which implies that $f^{-1}(V)$ is open.

(ii) $\Longrightarrow$ (i): Suppose $f^{-1}(V)$ is open for every open $V \subseteq Y$. Let $x \in X$ be arbitrary. Let us show that f is continuous at point x. Let $f(x) \in V$ for open $V \subseteq Y$. Then $U = f^{-1}(V)$ is an open set that satisfies $x \in U$ and $f(U) \subseteq V$. So f is continuous at x.

(ii) $\Longleftrightarrow$ (iii): Follows directly from the fact that for every $A \subseteq Y$ holds

$$X \setminus f^{-1}(A) = f^{-1}(Y \setminus A).$$

□

The following proposition characterizes continuous functions from a metric space:

Proposition 72 *Let X be a metric space and Y a topological space. Then $f : X \longrightarrow Y$ is continuous if and only if for every converging sequence $x_1, x_2, \ldots$ the sequence $f(x_1), f(x_2), \ldots$ converges and*

$$\lim_{i\to\infty} f(x_i) = f(\lim_{i\to\infty} x_i).$$

Proof. "$\Longrightarrow$": Suppose that f is continuous and let $x_1, x_2, \ldots$ be a converging sequence of elements of X. Let $x = \lim_{i\to\infty} x_i$. Let us prove that $f(x_1), f(x_2), \ldots$ converges to $f(x)$. Let U be an open set that contains $f(x)$. Then $f^{-1}(U)$ is open and $x \in f^{-1}(U)$. Because $x_1, x_2, \ldots$ converges to x there is n such that $x_i \in f^{-1}(U)$ for all $i \geq n$. But then $f(x_i) \in U$ for all $i \geq n$.

"$\Longleftarrow$": Let $x \in X$. To prove that f is continuous at point x, we assume the contrary and derive a contradiction. So suppose there is an open $V \subseteq Y$ that contains $f(x)$ such that for every δ there is a point y in $B_\delta(x)$ such that $f(y) \notin V$. Using $\delta = \frac{1}{i}$ for positive integers i we see that for every i there is $x_i \in X$ such that $d(x_i, x) < \frac{1}{i}$ and $f(x_i) \notin V$. Sequence $x_1, x_2, \ldots$ converges to x so by the hypothesis sequence $f(x_1), f(x_2), \ldots$ converges to $f(x) \in V$. But this is not possible since all $f(x_i) \notin V$. □

Observe the similarity of the condition in the previous proposition with Proposition 5 that deals with cellular automata. This is not coincidental: Later we'll use Proposition 5 and Proposition 72 to conclude that G is a continuous function.

Next propositions give some properties of continuous functions and compact sets.

Proposition 73 *Suppose function $f : X \longrightarrow Y$ is continuous. For every compact A the set $f(A)$ is compact.*

Proof. Consider an open cover of $f(A)$ by open sets V_i. Then, by Proposition 71 the sets $f^{-1}(V_i)$ form an open cover of A. By compactness of A there is a finite subcover of A by $f^{-1}(V_i)$ where $i \in F$ for some finite set F. But then the corresponding sets V_i for $i \in F$ form a finite subcover of $f(A)$. Hence $f(A)$ is compact. □

Proposition 74 *If $f : X \longrightarrow Y$ is a continuous bijection where X is compact and Y is Hausdorff then the inverse function $f^{-1} : Y \longrightarrow X$ is also continuous.*

Proof. By Proposition 71 it is enough to show that for every closed $A \subseteq X$ also $f(A)$ is closed. But if $A \subseteq X$ is closed then by Proposition 65 it is also compact. By Proposition 73 set $f(A)$ is also compact, and then by Proposition 66 set $f(A)$ is closed. □

5.2 Basic facts about the configuration space

Let us return to the space of interest to us: The space $S^{\mathbb{Z}^d}$ of configurations, with the metric

$$d(e,c) = \begin{cases} 0, & \text{if } e = c, \\ 2^{-\min\{\|\vec{x}\| \ \mid \ c(\vec{x}) \neq e(\vec{x})\}}, & \text{if } e \neq c. \end{cases}$$

The open ball of radius $\varepsilon = 2^{-r}$ centered at $c \in S^{\mathbb{Z}^d}$ is

$$B_\varepsilon(c) = \{e \in S^{\mathbb{Z}^d} \mid e(\vec{x}) = c(\vec{x}) \text{ for all } \|\vec{x}\| \leq r\}.$$

These balls hence form a base. This already implies that the topology has a countable base.

More generally, for any finite domain $D \subseteq \mathbb{Z}^d$ and configuration $c \in S^{\mathbb{Z}^d}$ we define the *cylinder set*

$$\mathrm{Cyl}(c, D) = \{e \in S^{\mathbb{Z}^d} \mid e(\vec{x}) = c(\vec{x}) \text{ for all } \vec{x} \in D\}$$

that contains all those configurations that agree with c in domain D.

Note that for sufficiently large r we have $D \subseteq E$ where

$$E = \{\vec{x} \in \mathbb{Z}^d \mid \|\vec{x}\| \leq r\}.$$

Then

$$\mathrm{Cyl}(c, D) = \bigcup_{e \in \mathrm{Cyl}(c,D)} \mathrm{Cyl}(e, E),$$

so all cylinders are (finite) unions of open balls, and hence they are open in the topology. Balls form a base of the topology, so also cylinders form a base.

Let us next show that a sequence of configurations $c_1, c_2, \ldots$ converges to $c \in S^{\mathbb{Z}^d}$ in this topology, if and only if it converges to c according to the definition of convergence in Section 1.8. First, suppose convergence to c in the topology, and let $\vec{n} \in \mathbb{Z}^d$ be arbitrary. Consider the cylinder U determined by c and domain $\{\vec{n}\}$. Convergence to c implies that for all sufficiently large i holds $c_i \in U$, that is, $c_i(\vec{n}) = c(\vec{n})$. So the sequence converges to c according to the definition of Section 1.8.

Conversely, suppose converge to c as defined in Section 1.8. Let U be an open set that contains c. Because cylinders form a base, there is a finite $D \subseteq \mathbb{Z}^d$ such that $\mathrm{Cyl}(c, D) \subseteq U$. By the definition of convergence of $c_1, c_2, \ldots$ there is $k \in \mathbb{Z}$ such that $c_i \in \mathrm{Cyl}(c, D)$ for all $i > k$. This means that the sequence converges to c in the topology.

Now we immediately obtain the following corollaries of our earlier propositions:

Corollary 75 *Metric space $S^{\mathbb{Z}^d}$ is compact.*

Proof. Follows directly from Propositions 4 and 64. □

Since $S^{\mathbb{Z}^d}$ is metric and compact, all propositions of Section 5.1 hold in this set-up. In particular, $S^{\mathbb{Z}^d}$ is Hausdorff, complete, separable, second countable and a Baire space.

Corollary 76 *Every CA function $G : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$ is continuous.*

Proof. Follows directly from Propositions 5 and 72. □

Pairs (X, F) where X is compact and $F : X \longrightarrow X$ is continuous are commonly called (topological) dynamical systems. So we see that $(S^{\mathbb{Z}^d}, G)$ is a (topological) dynamical system for each CA function G.

5.3 Hedlund's theorem

We know that every CA function is continuous and commutes with translations. The next result by Hedlund states that also the converse is true: Any function $S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$ that is continuous and commutes with translations is a CA function. As we see in the proof below, continuity implies that there is a local rule for each cell, and translation invariance implies that the local rules are the same for all cells.

Proposition 77 *Function $G : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$ is a CA function if and only if it is continuous and it commutes with translations of $\mathbb{Z}^d$.*

Proof. One direction follows from Corollary 76 and Proposition 3. For the other direction, suppose that G is continuous and commutes with translations. It was proved in the homework assignments that in compact spaces continuous functions are uniformly continuous. In particular, this implies that there exists $r > 0$ such that

$$d(c, e) < 2^{-r} \Longrightarrow d(G(c), G(e)) < 1,$$

or, equivalently,

$$c(\vec{x}) = e(\vec{x}) \text{ for all } \|\vec{x}\| \leq r \implies G(c)(\vec{0}) = G(e)(\vec{0}).$$

This provides a radius-r CA function F such that for all $c \in S^{\mathbb{Z}^d}$ holds $G(c)(\vec{0}) = F(c)(\vec{0})$.

It was assumed that G commutes with translations, and we know that CA function F commutes with translations. Hence, for every $\vec{n} \in \mathbb{Z}^d$ and every $c \in S^{\mathbb{Z}^d}$

$$[G(c)]\,(\vec{n}) = [\tau(G(c))]\,(\vec{0}) = [G(\tau(c))]\,(\vec{0}) = [F(\tau(c))]\,(\vec{0}) = [\tau(F(c))]\,(\vec{0}) = [F(c)]\,(\vec{n})$$

where we denoted by τ the translation determined by $\vec{n}$. We conclude that $G = F$, so G is a CA function. □

Note how we now get a direct topological proof of Proposition 9: If CA function $G : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$ is bijective then by Proposition 74 the inverse function G^{-1} is continuous. Since trivially G^{-1} commutes with translations ($\tau \circ G^{-1} = G^{-1} \circ G \circ \tau \circ G^{-1} = G^{-1} \circ \tau \circ G \circ G^{-1} = G^{-1} \circ \tau$ for any translation τ), it follows from Hedlund's theorem that G^{-1} is a CA function.

5.4 Limit sets and attractors

The dynamics of a CA on Garden-of-Eden configurations is not relevant when analyzing its long-term behavior. They only represent a transient part in the dynamics. So it makes sense to restrict the attention to points that are not Garden-of-Eden for G, or for G^n for any finite n. This relevant part of the space is the limit set of the system.

More formally, the limit set Ω_G of CA G is the intersection of all forward images of $S^{\mathbb{Z}^d}$:

$$\Omega_G = \bigcap_{n=0}^{\infty} G^n(S^{\mathbb{Z}^d}).$$

This means that $c \in \Omega_G$ if and only if c is not a Garden-of-Eden of G^n for any n.

Example 38. The limit set of the *xor* CA of Example 1 is the whole configuration space $\{0,1\}^{\mathbb{Z}}$. This is of course true for any surjective CA. Consider then a nilpotent CA, i.e. a CA G in which every configuration eventually evolves into the quiescent configuration c_q. It was proved in the beginning of Section 3.4 that there is a number n such that G^n maps every configuration into c_q. This means that $G^k(S^{\mathbb{Z}^d}) = \{c_q\}$ for all $k \geq n$, so $\Omega_G = \{c_q\}$.

As a non-surjective, non-nilpotent example consider the elementary CA with Wolfram number 128. It is a CA in which the quiescent state 0 is spreading: every cell whose neighborhood contains 0 becomes 0, and pattern 111 is the only one that is mapped into 1 by the local rule. In this CA every finite configuration eventually becomes the quiescent configuration, but the CA is not nilpotent since, for example, configuration $\ldots 111 \ldots$ is a fixed point. Note that every pattern $100\ldots01$ is an orphan for G^n for sufficiently large n, so Ω_G does not contain any configuration with such a pattern. This means that the only potential elements of Ω_G are configurations where 1's form a contiguous segment. These are configurations

$$\begin{array}{c}\ldots 111111 \ldots\\ \ldots 000000 \ldots\\ \ldots 000111 \ldots\\ \ldots 111000 \ldots\\ \ldots 000111 \ldots 111000 \ldots\end{array}$$

It is easy to see that each such configuration has a pre-image of the same form, which implies that they all belong to the limit set. We see that in this case the limit set is countably infinite. □

The following proposition states a few simple but fundamental properties of limit sets. A *subshift* of $S^{\mathbb{Z}^d}$ is any $X \subseteq S^{\mathbb{Z}^d}$ that is topologically closed, non-empty and translation invariant (i.e. $\tau(X) = X$ for all translations τ). The whole configuration space $S^{\mathbb{Z}^d}$ is also called a d-dimensional *full shift.*

Proposition 78 *Let G be a CA function and let Ω be its limit set. Then*

(a) Ω is a subshift.

(b) The limit set satisfies $G(\Omega) = \Omega$, and if $X \subseteq S^{\mathbb{Z}^d}$ satisfies $G(X) = X$ then $X \subseteq \Omega$.

(c) For every configuration c we have that $c \in \Omega$ if and only if there is a sequence $\ldots, c_{-2}, c_{-1}, c_0$ of configurations such that $c_0 = c$ and $G(c_i) = c_{i+1}$ for all $i < 0$. (In other words, elements of Ω are exactly the configurations that belong to two-way infinite orbits.)

(d) The limit set is finite if and only if the CA is nilpotent, in which case the limit set contains just one configuration. (See the beginning of Section 3.4 for the definition of nilpotent CA.)

Proof. (a) Because G^n is continuous for every n, it follows from Proposition 73 that $G^n(S^{\mathbb{Z}^d})$ is compact for every n. Hence it is closed (Proposition 66) which means that the limit set Ω is closed (Proposition 57). Consequently, Ω is compact (Proposition 65). Note that

$$S^{\mathbb{Z}^d} \supseteq G(S^{\mathbb{Z}^d}) \supseteq G^2(S^{\mathbb{Z}^d}) \supseteq G^3(S^{\mathbb{Z}^d}) \supseteq \ldots$$

forms a descending chain of non-empty compact sets. It follows then from the finite intersection property (Proposition 62) that their intersection Ω cannot be empty. (Another, non-topological proof for non-emptiness of Ω is based on the simple observation that some homogeneous configuration $\ldots aaa \ldots$ must be temporally periodic, and hence in Ω.)

To prove translation invariance, consider an arbitrary translation τ and arbitrary $c \in \Omega$. For every n, we have $c = G^n(e)$ for some $e \in S^{\mathbb{Z}^d}$, so

$$\tau(c) = \tau(G^n(e)) = G^n(\tau(e))$$

which means that $\tau(c) \in G^n(S^{\mathbb{Z}^d})$. This is true for all n, so $\tau(c) \in \Omega$. We conclude that $\tau(\Omega) \subseteq \Omega$. Then also $\tau^{-1}(\Omega) \subseteq \Omega$ so that $\Omega = \tau(\tau^{-1}(\Omega)) \subseteq \tau(\Omega)$. We conclude that $\Omega = \tau(\Omega)$.

(b) Always $G^{-n}(c) \subseteq G^{-n-1}(G(c))$ so if $c \in \Omega$ then also $G(c) \in \Omega$. Hence $G(\Omega) \subseteq \Omega$. To prove $\Omega \subseteq G(\Omega)$ consider an arbitrary $c \in \Omega$. Because $\{c\}$ is closed and G is continuous we have that $G^{-1}(c)$ is closed. For every n the set $G^n(S^{\mathbb{Z}^d})$ is also closed and hence the intersection

$$A_n = G^{-1}(c) \cap G^n(S^{\mathbb{Z}^d})$$

is closed, thus compact. The intersection is also non-empty because $G^{-n-1}(c)$ is not empty. This means that

$$A_1 \supseteq A_2 \supseteq A_3 \supseteq \ldots$$

is a decreasing chain of non-empty compact sets. It follows from the finite intersection property (Proposition 62) that

$$\bigcap_{n=1}^{\infty} A_n \neq \emptyset.$$

Let $e \in A_n$ for all n, so $G(e) = c$ and $e \in G^n(S^{\mathbb{Z}^d})$ for all n. We see that c has a pre-image e that belongs to the limit set Ω, so $c \in G(\Omega)$, as required.

For the second part of (b), suppose X satisfies $G(X) = X$. Then $G^n(X) = X$ for all n, and therefore $X \subseteq G^n(S^{\mathbb{Z}^d})$ for all n. Thus $X \subseteq \Omega$.

(c) If c belongs to some two-way infinite orbit then $G^{-n}(c)$ is not empty for any n, so $c \in \Omega$. Conversely, suppose that $c_0 \in \Omega$. It follows from (b) that there exists $c_{-1} \in \Omega$ such that $G(c_{-1}) = c_0$. The same argument for c_{-1} yields c_{-2} that satisfies $G(c_{-2}) = c_{-1}$, and by iterating the reasoning we obtain the required sequence $\ldots, c_{-2}, c_{-1}, c_0$.

(d) If G is nilpotent then Ω contains only the quiescent configuration c_q. Suppose then that G is not nilpotent. Let σ be the translation determined by vector $\vec{e_1} = (1, 0, 0, \ldots, 0)$. Let us prove that Ω contains a configuration c such that $\sigma^n(c) \neq c$ for all $n \in \mathbb{Z}$. Then, by (a), we have $\sigma^n(c) \in \Omega$ for all n, which means that there are infinitely many different elements in Ω.

Let $c_q \in \Omega$ be a homogeneous configuration in the limit set where every cell is in state q. It exists for some state q. Since G is not nilpotent there exists state $a \in S$ that is different from q such that for every n there is a configuration e_n such that $G^n(e_n)$ has some cell in state a. Combining a configuration that produces in n steps c_q and configuration e_n provides a configuration whose image under G^n has all cells $(x_1, x_2, \ldots, x_d)$ with $x_1 > 0$ in state q, and some cell is in state a. Translating suitably we see that there is a configuration f_n for every n such that $G^n(f_n)$ has state $\neq q$ in cell $\vec{0}$ and state q in all cells $(x_1, x_2, \ldots, x_d)$ with $x_1 > 0$. Let

$$B = \{c \in S^{\mathbb{Z}^d} \mid c(\vec{0}) \neq q \text{ and } c(x_1, x_2, \ldots, x_d) = q \text{ when } x_1 > 0\ \}.$$

Set B is topologically closed and, as shown above, $B \cap G^n(S^{\mathbb{Z}^d}) \neq \emptyset$ for all n. It follows from the finite intersection property that $B \cap \Omega \neq \emptyset$. All $c \in B \cap \Omega$ satisfy $\sigma^k(c) \neq c$ so we have found a configuration with the desired properties. □

The limit set is an example of an attractor of a CA. (In fact it is the unique maximal attractor.) A closed set C is called *inward* for CA G if $G(C) \subseteq C^\circ$, where C° is the interior of C. A clopen set U is then inward iff $G(U) \subseteq U$. Non-empty set $A \subseteq S^{\mathbb{Z}^d}$ is called an *attractor* for CA G if

$$A = \bigcap_{n=0}^{\infty} G^n(U)$$

for some inward clopen $U \subseteq S^{\mathbb{Z}^d}$. We call A the attractor determined by U. It is clear that the limit set Ω is an attractor (determined by $U = S^{\mathbb{Z}^d}$), and that every attractor is a subset of Ω. It is also clear that every attractor is compact and contains a totally periodic configuration. If A is an attractor then $G(A) = A$ which can be seen in the same way as $G(\Omega) = \Omega$ was proved in Proposition 78(b). Finally, notice that since there are only countably many different clopen sets, the number of attractors is countable. (Remark: a set is clopen iff it is a finite union of cylinders.)

We begin with some general properties of attractors that, in fact, hold in any topological dynamical system.

Proposition 79 *Let A and B be two attractors of G.*

(a) The union $A \cup B$ is an attractor.

(b) If $A \cap B \neq \emptyset$ then there exists an attractor $C \subseteq A \cap B$.

Proof. Let U, V be inward clopen sets that specify the attractors A and B, respectively.

(a) The union $U \cup V$ is clopen and satisfies $G(U \cup V) \subseteq U \cup V$ so it determines an attractor

$$C = \bigcap_{n=0}^{\infty} G^n(U \cup V) = \bigcap_{n=0}^{\infty} G^n(U) \cup G^n(V).$$

It is clear that $A \subseteq C$ and $B \subseteq C$ so $A \cup B \subseteq C$. For the converse inclusion, let $c \in C$ be arbitrary. Then, for every $n = 0, 1, 2, \ldots$ we have $c \in G^n(U)$ or $c \in G^n(V)$. Therefore, $c \in G^n(U)$ for infinitely many n, or $c \in G^n(V)$ for infinitely many n. But because

$$U \supseteq G(U) \supseteq G^2(U) \supseteq \ldots,$$

it follows that if $c \in G^n(U)$ for infinitely many n then $c \in G^n(U)$ for all n. Analogous statement holds in the case that $c \in G^n(V)$ for infinitely many n. So we have that $c \in A$ or $c \in B$.

(b) If $A \cap B \neq \emptyset$ we must have $U \cap V \neq \emptyset$. The intersection $U \cap V$ is clopen and inward so it determines an attractor C. We clearly have $C \subseteq A$ and $C \subseteq B$. □

Attractor A is called *minimal* if there are no proper subsets that are attractors. If A is an attractor then its *basin of attraction* is the set

$$\mathcal{B}_A = \{c \in S^{\mathbb{Z}^d} \mid \lim_{n\to\infty} d(G^n(c), A) = 0\}$$

where

$$d(x, A) = \min\{d(x, a) \mid a \in A\}.$$

(Minimum exists due to the compactness of A.) In other words, the basin of attraction consists of all points that are attracted to A in the sense that they approach A in the limit.

Proposition 80 *Let A be an attractor determined by clopen U that satisfies $G(U) \subseteq U$, and let $\mathcal{B}$ be the basin of attraction for A. Then*

(a) $G(\mathcal{B}) \subseteq \mathcal{B}$ and $G(S^{\mathbb{Z}^d} \setminus \mathcal{B}) \subseteq S^{\mathbb{Z}^d} \setminus \mathcal{B}$,

(b) $U \subseteq \mathcal{B}$ so that, in particular, for all $c \in S^{\mathbb{Z}^d}$,

$$\lim_{n\to\infty} d(G^n(c), \Omega_G) = 0,$$

(c) $\mathcal{B} = \bigcup_{n=0}^{\infty} G^{-n}(U)$*,*

(d) $\mathcal{B}$ *is open,*

(e) $\mathcal{B}$ *identifies* A *uniquely, that is, for attractors* A_1, A_2 *holds* $A_1 \neq A_2 \Longrightarrow \mathcal{B}_{A_1} \neq \mathcal{B}_{A_2}$*.*

Proof. (a) If $c \in \mathcal{B}$ then $\lim_{n\to\infty} d(G^n(G(c)), A) = \lim_{n\to\infty} d(G^{n+1}(c), A) = 0$ so $G(c) \in \mathcal{B}$. Conversely, if $G(c) \in \mathcal{B}$ then $\lim_{n\to\infty} d(G^n(c), A) = \lim_{n\to\infty} d(G^{n-1}(G(c)), A) = 0$, so $c \in \mathcal{B}$.

(b) For every $\varepsilon > 0$ let

$$B_\varepsilon(A) = \bigcup_{x \in A} B_\varepsilon(x).$$

Set $B_\varepsilon(A)$ is open so its complement is closed. Let us prove that there is n such that $G^n(U) \subseteq B_\varepsilon(A)$. If not, then for every n the intersection of $G^n(U)$ and the complement of $B_\varepsilon(A)$ is a non-empty compact set. Let C_n be this non-empty intersection. Clearly

$$C_1 \supseteq C_2 \supseteq C_3 \supseteq \dots$$

so it follows from the finite intersection property of compact sets that

$$\bigcap_{i=1}^{\infty} C_i \neq \emptyset.$$

But his means that there is a configuration in A that is not in $B_\varepsilon(A)$, a contradiction.

Because $U \supseteq G(U) \supseteq G^2(U) \supseteq \dots$ we see that $G^n(U) \subseteq B_\varepsilon(A)$ for all sufficiently large n. Let $c \in U$ be arbitrary. Then for every ε we have $d(G^n(c), A) < \varepsilon$ for all sufficiently large n. This is true for every positive ε, so $c \in \mathcal{B}$.

(c) It follows from (a) and (b) that $G^{-n}(U) \subseteq \mathcal{B}$ for all n, so the inclusion from right to left is clear. For the converse inclusion we observe first that for every cylinder set C there exists $\varepsilon > 0$ such that $B_\varepsilon(C) = C$. Clopen sets are finite unions of cylinders, so for clopen U holds $B_\varepsilon(U) = U$ for some $\varepsilon > 0$. Because $A \subseteq U$

$$B_\varepsilon(A) \subseteq B_\varepsilon(U) = U.$$

If $c \in \mathcal{B}$ then $G^n(c) \in B_\varepsilon(A)$ for sufficiently large n, so $G^n(c) \in U$. This means that $c \in G^{-n}(U)$.

(d) Follows directly from (c) and the facts that U is open and G is continuous.

(e) Let A_1 and A_2 be attractors determined by inward clopen U_1 and U_2, respectively. Let $\mathcal{B}$ be the basin of attraction for both A_1 and A_2. By (b), $U_1 \subseteq \mathcal{B}$, so by (c) we have $U_1 \subseteq \bigcup_{n=0}^{\infty} G^{-n}(U_2)$. Each $G^{-n}(U_2)$ is open, so by compactness of U_1 we have that $U_1 \subseteq G^{-m}(U_2)$ for some m. This simply means that $G^m(U_1) \subseteq U_2$, so that

$$A_1 = \bigcap_{n=0}^{\infty} G^n(U_1) \subseteq \bigcap_{n=0}^{\infty} G^n(U_2) = A_2.$$

Analogously $A_2 \subseteq A_1$ so that $A_1 = A_2$. □

Example 39. Consider the *xor* CA. Let us prove that the limit set is the only attractor. For any finite word $w \in \{0,1\}^*$ and $n \in \mathbb{Z}$ let us denote by

$$\text{Cyl}(w,n) = \{c \in \{0,1\}^{\mathbb{Z}} \mid c(n)c(n+1)\dots c(n+|w|-1) = w\}$$

the cylinder of configurations that contain pattern w starting in position n. For non-empty w let us denote by $G(w)$ the word u of length $|w|-1$ where $u_i \equiv w_i + w_{i+1} \pmod 2$. It is easy to see that

$$G(\text{Cyl}(w,n)) = \text{Cyl}(G(w),n).$$

Suppose A is an attractor determined by clopen U that satisfies $G(U) \subseteq U$. Let w be the shortest word such that $\text{Cyl}(w,n) \subseteq U$ for some $n \in \mathbb{Z}$. If w is the empty word then $U = \{0,1\}^{\mathbb{Z}}$ so A is the limit set $\{0,1\}^{\mathbb{Z}}$. If w is not the empty word then

$$\text{Cyl}(G(w),n) = G(\text{Cyl}(w,n)) \subseteq G(U) \subseteq U,$$

which contradicts the choice of w since $G(w)$ is shorter than w. We conclude that $U = \{0,1\}^{\mathbb{Z}}$ is the only clopen set that satisfies $G(U) \subseteq U$, and the limit set is the only attractor. □

Example 40. Consider the elementary rule 128. We found its limit set in Example 38. It has another attractor that contains only one configuration $\dots 000 \dots$. This attractor is determined by the clopen cylinder

$$U = \{c \mid c(0) = 0\}.$$

The basin $\mathcal{B}$ of attraction consists of all configurations in which cell 0 eventually becomes 0, which means that all configurations except $\dots 111 \dots$ are in $\mathcal{B}$.

There are no other attractors: If X is clopen then X contains a configuration with some cell in state 0. This implies that the attractor determined by X necessarily contains configuration $\dots 000 \dots$. But then its basin of attraction contains all configurations with some cell in state 0. There are only two such sets, and they are the basins of attraction of the two attractors above, so no other attractors exist. □

The following proposition discusses the possible attractor structures of cellular automata. There two basic alternatives: Either there are two disjoint attractors, in which case there in fact are infinitely many disjoint attractors and a hierarchy of nested attractors, or the intersection of all attractors is non-empty.

Proposition 81 *For every CA G either*

(a) there are two attractors whose intersection is empty, or

(b) the intersection of all attractors is non-empty.

Proof. Suppose there are no disjoint attractors, and let us prove that the intersection of all attractors is non-empty.

Let $U_1, U_2, \ldots$ be a sequence that contains all clopen sets that satisfy $G(U_i) \subseteq U_i$. Then each attractor is determined by some U_i. Suppose there is n such that $U_1 \cap U_2 \cap \ldots \cap U_n$ is empty. Let n be smallest such number, so that $U = U_1 \cap U_2 \cap \ldots \cap U_{n-1}$ is not empty but $U \cap U_n$ is empty. Finite intersections of clopen sets are clopen so U is clopen. Since $G(U_i) \subseteq U_i$ for all i, we have $G(U) \subseteq U$. Hence there are clopen sets U and U_n that are disjoint and satisfy $G(U) \subseteq U$ and $G(U_n) \subseteq U_n$. The attractors specifies by U and U_n are disjoint and condition (a) of the proposition holds.

Suppose then that for every n the intersection $U_1 \cap U_2 \cap \ldots \cap U_n$ is non-empty. It follows from the finite intersection property of compact sets that the intersection V of all U_i's is non-empty. Set V is closed (but not necessarily open) and it satisfies $G(V) \subseteq V$ and $V \subseteq U_i$ for all i. The intersection

$$\bigcap_{n=0}^{\infty} G^n(V)$$

is non-empty (finite intersection property) and it is contained in every attractor. Hence the intersection of attractors is non-empty. □

In case (b) of the previous proposition the intersection of attractors can itself be an attractor (in which case it is the unique minimal attractor). As seen in Example 40, elementary CA 128 has such intersection. In some cases the intersection of all attractors is not an attractor, as shown by the following example.

Example 41. Consider elementary CA 136. In this CA the new state of a cell is 1 if and only if the old states of the cell and its right neighbor were 1. Every clopen set contains a finite configuration, and all finite configurations evolve into the quiescent configuration $\ldots 0000 \ldots$. This means that the quiescent configuration is in every attractor. So condition (b) of Proposition 81 holds. For every positive integer m we denote by C_m the cylinder of all configurations that have state 0 in cells $1, 2, \ldots m$. Since $G(C_m) \subseteq C_m$, cylinder C_m determines an attractor A_m. Since state 0 spreads to the left, all configurations in A_m have state 0 in all cells $i \leq m$. So the intersection of all attractors A_m is the singleton $A = \{\ldots 0000 \ldots\}$.

However, the intersection A of all attractors is not an attractor itself: Every clopen set U contains a configuration where cells j are in state 1 for all sufficiently large j, and therefore the corresponding attractor also contains such a configuration. Hence the attractor determined by U is different from A. □

The following is an example of type (a) attractor structure.

Example 42. Consider the majority CA (elementary CA 232, see Example 6). The cylinder C_{00} determined by pattern 00 satisfies $G(C_{00}) \subseteq C_{00}$. Also the cylinder C_{11} determined by pattern 11 satisfies $G(C_{11}) \subseteq C_{11}$. Both cylinders determine an attractor, and since the cylinders are disjoint the attractors they determine are also disjoint. □

The following proposition analyzes type (a) attractor structure. In this case, it turns out that no minimal attractors exist:

Proposition 82 *If there exist two disjoint attractors then every attractor contains as subset two disjoint attractors.*

Proof. Let A and B be two disjoint attractors, determined by clopen sets U and V, respectively. Intersection $U \cap V$ is clopen and satisfies $G(U \cap V) \subseteq U \cap V$. If $U \cap V$ is not empty then

$$\bigcap_{n=0}^{\infty} G^n(U \cap V)$$

is a non-empty set that is a subset of both A and B, which contradicts the fact that A and B are disjoint. We conclude that U and V must be disjoint.

Let C be an arbitrary attractor, determined by clopen set W. There is a translation τ such that $U' = \tau(U) \cap W$ and $V' = \tau(V) \cap W$ are non-empty. Note that U' and V' are clopen, $U' \cap V' = \emptyset$ and $U', V' \subseteq W$. Moreover, $G(U') \subseteq U'$ and $G(V') \subseteq V'$. All this means that the attractors determined by U' and V' are disjoint subsets of C. □

In particular, it follows from the previous proposition that if two disjoint attractors exist then there exist infinitely many attractors that are pairwise disjoint. Indeed, if A_1 and B_1 are disjoint attractors then B_1 contains two disjoint attractors A_2 and B_2, B_2 contains two disjoint attractors A_3 and B_3, and so on. Sets $A_1, A_2, A_3, \ldots$ are pairwise disjoint attractors of the CA.

It also follows from the proposition that if disjoint attractors exist then there is no minimal attractor since every attractor contains proper subsets that are attractors.

Proposition 83 *A minimal attractor of a CA is always the intersection of all attractors. In particular: Every CA has at most one minimal attractor.*

Proof. Let A be a minimal attractor, and let B be an arbitrary attractor. If A and B are disjoint then by Proposition 82 attractor A contains proper subsets that are attractors, which contradicts the minimality of A. So $A \cap B \neq \emptyset$. Since $A \cap B$ contains an attractor (determined by $U \cap V$ where U and V are clopen sets that determine A and B, respectively) it follows from the minimality of A that this attractor must be equal to A. Hence $A \subseteq B$. □

In summary: a minimal attractor exists if and only if the intersection of all attractors is an attractor. This intersection is then the unique minimal attractor. Other possibilities are that there are disjoint attractors, or that the intersection of all attractors is a non-empty set that is not an attractor. In these cases no minimal attractor exists.

5.5 Equicontinuity and sensitivity

In this section we study concepts that are related to the propagation of changes in the initial configuration. Loosely speaking, a dynamical system is "chaotic" if small perturbations in the state of the system get magnified during the evolution. This means that we cannot predict the long term behavior of the system by simulation unless the initial state is known precisely. In contrast, configurations whose orbits are arbitrarily well tracked by all sufficiently close configurations are called equicontinuous. Such orbits can be reliably simulated.

Precisely speaking, configuration $c \in S^{\mathbb{Z}^d}$ is an *equicontinuity point* for CA G if

$$\forall \varepsilon > 0,\ \exists \delta > 0,\ \forall e \in B_\delta(c),\ \forall n \geq 1: \quad d(G^n(c), G^n(e)) < \varepsilon.$$

In other words, all functions $G, G^2, G^3, \ldots$ are continuous at point c with the same positive parameter $\delta > 0$ corresponding to any $\varepsilon > 0$. (Hence the term "equicontinuous".) Let $\mathcal{E} = \mathcal{E}_G$ denote the set of equicontinuity points of G.

In terms of cylinders, the equicontinuity property can be rephrased as follows: $c \in \mathcal{E}$ if and only if for every finite observation window $E \subseteq \mathbb{Z}^d$ there corresponds a finite domain $D \subseteq \mathbb{Z}^d$ such that

$$e \in \mathrm{Cyl}(c, D) \Longrightarrow \forall n \geq 1 : G^n(e) \in \mathrm{Cyl}(G^n(c), E).$$

Cellular automaton G is called *equicontinuous* if all configurations are equicontinuous, that is, if $\mathcal{E} = S^{\mathbb{Z}^d}$. It turns out that only relatively trivial (namely eventually periodic) CA are equicontinuous.

Proposition 84 *A CA G is equicontinuous if and only if $G^{m+p} = G^m$ for some $p \geq 1$ and $m \geq 0$.*

Proof. If $G^{m+p} = G^m$ then there are only finitely many different functions $G, G^2, G^3, \ldots$, so the equicontinuity of G follows directly.

For the other direction, suppose that G is equicontinuous. In the definition of equicontinuity, choose $\varepsilon = 1$. We see that every $c \in S^{\mathbb{Z}^d}$ has an open neighborhood U_c such that

$$e \in U_c \Longrightarrow d(G^n(c), G^n(e)) < 1, \text{ for all } n \geq 1.$$

By compactness, there are finitely many configurations $c_1, c_2, \ldots, c_n$ such that $U_{c_1}, U_{c_2}, \ldots, U_{c_n}$ is a cover of $S^{\mathbb{Z}^d}$. This means that for every $e \in S^{\mathbb{Z}^d}$ there exists $i \in \{1, 2, \ldots, n\}$ such that $G^n(e)(\vec{0}) = G^n(c_i)(\vec{0})$ for all n.

Totally (spatially) periodic configurations are dense in $S^{\mathbb{Z}^d}$, so we can choose $c_1, c_2, \ldots, c_n$ to be totally periodic. Because totally (spatially) periodic configurations are (temporally) eventually periodic, for each $i \in \{1, 2, \ldots, n\}$ there exist $m_i \geq 0$ and $p_i \geq 1$ such that $G^{m_i+p_i}(c_i) = G^{m_i}(c_i)$. If we choose $m \geq m_1, m_2, \ldots m_n$ and $p = p_1 p_2 \ldots p_n$, then $G^{m+p}(c_i) = G^m(c_i)$ for all i, so that

$$G^{m+p}(e)(\vec{0}) = G^{m+p}(c_i)(\vec{0}) = G^m(c_i)(\vec{0}) = G^m(e)(\vec{0})$$

for all $e \in S^{\mathbb{Z}^d}$. By translation invariance of G, we have $G^{m+p} = G^m$. □

Corollary 85 *A surjective CA G is equicontinuous if and only if $G^p = id$ for some $p \geq 1$.*

Proof. Suppose G is surjective and equicontinuous. By Proposition 84 there exist $p \geq 1$ and $m \geq 0$ such that $G^{m+p} = G^m$. Let $c \in S^{\mathbb{Z}^d}$ be arbitrary. By surjectivity there exists $e \in S^{\mathbb{Z}^d}$ such that $G^m(e) = c$. Hence $G^p(c) = G^{m+p}(e) = G^m(e) = c$, so we see that $G^p = id$. The other direction is trivial. □

Cellular automaton G is *sensitive to initial conditions* (or simply *sensitive*) if there is a positive number $\varepsilon > 0$ such that for every configuration c there exist configurations e arbitrarily close to c such that $d(G^n(c), G^n(e)) > \varepsilon$ for some n. More compactly stated: G is sensitive iff

$$\exists \varepsilon > 0,\ \forall c \in S^{\mathbb{Z}^d},\ \forall \delta > 0,\ \exists e \in B_\delta(c),\ \exists n \geq 1 : \quad d(G^n(c), G^n(e)) > \varepsilon.$$

In cellular automata terminology this means that there is a finite observation window $E \subseteq \mathbb{Z}^d$ such that for every configuration c and every domain D, there exists a configuration $e \in \mathrm{Cyl}(c, D)$ such that $G^n(e) \notin \mathrm{Cyl}(G^n(c), E)$, for some $n \geq 1$.

Example 43. Any non-zero translation τ is sensitive to initial conditions. As another example, consider the one-dimensional *xor* CA. For any configuration c and positive integer n the configuration e obtained from c by toggling the state of cell n has the property that $d(c, e) = 2^{-n}$ while $d(G^n(c), G^n(e)) = 1$. Hence *xor* is sensitive. Note also that if G is sensitive and H is an arbitrary CA then the product $G \times H$ is sensitive, where the product CA consists of two tracks where G and H act independently of each other. In particular this means that non-surjective CA can be sensitive. □

It is easy to see that a sensitive CA cannot have any equicontinuity points.

Proposition 86 *If G is sensitive then $\mathcal{E}_G = \emptyset$.*

Proof. If G is sensitive then by changing the order of the first two quantifiers in the definition

$$\exists \varepsilon > 0,\ \forall c \in S^{\mathbb{Z}^d},\ \forall \delta > 0, \ldots$$

of sensitivity we obtain

$$\forall c \in S^{\mathbb{Z}^d},\ \exists \varepsilon > 0,\ \forall \delta > 0,\ \exists e \in B_\delta(c),\ \exists n \geq 1 : \quad d(G^n(c), G^n(e)) > \varepsilon.$$

This is equivalent to $\forall c \in S^{\mathbb{Z}^d} : c \notin \mathcal{E}_G$, that is, to $\mathcal{E}_G = \emptyset$. □

So far we discussed CA in arbitrary dimension d. Among one-dimensional CA one can prove further results (which are not valid in higher dimensions). In particular, (1) the converse of Proposition 86 holds (so a one-dimensional CA is sensitive if and only if it has no equicontinuity points), and (2) if a one-dimensional CA has any equicontinuity points then the set of equicontinuity points is dense (more precisely: $\mathcal{E}$ is then a residual set).

Recall that a set $A \subseteq S^{\mathbb{Z}^d}$ is called *residual* if it is the intersection of countably many dense open sets. By Proposition 70 the space $S^{\mathbb{Z}^d}$ is a Baire space, so any residual set is dense. Intuitively, residual sets are "fat" and contain "almost all" configurations.

The source of the differences between the one- and the higher dimensional equicontinuity points is the fact in the one-dimensional case equicontinuity is characterized by the presence of blocking words that prevent any information passing from one side to the other, while an analogous, two-dimensional counter part ("blocking rectangle") does not divide the space into disconnected areas, and information can circumvent the obstacle.

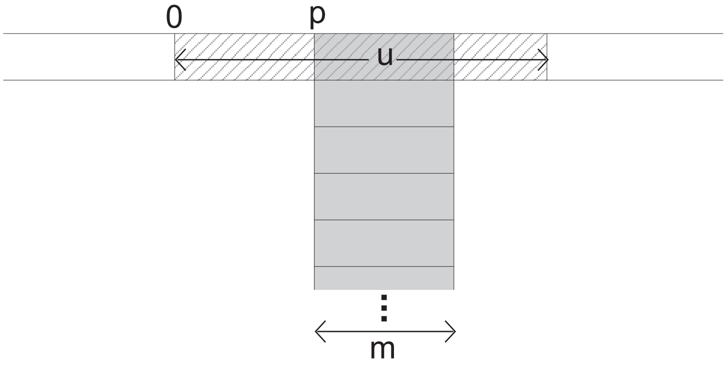
More precisely, consider a one-dimensional CA G over the state set S. An *m-blocking word u* is a word over the alphabet S such that for some offset $p \in \mathbb{Z}$

$$c, e \in \mathrm{Cyl}(u, 0) \implies G^n(c)(i) = G^n(e)(i) \text{ for all } n \geq 0 \text{ and } p \leq i < p + m.$$

Here we use the notation

$$\mathrm{Cyl}(u, n) = \{c \in S^{\mathbb{Z}} \mid c(n)c(n+1)\dots c(n+|u|-1) = u\}$$

for the cylinder of configurations that contain word u starting in position n. In other words, if c contains the blocking word u then the states of the cells in some segment of length m at all times are independent of the initial states outside of the blocking word u. In the following illustration, word u is blocking iff the states in the gray area are independent of the states outside of the shaded word u:



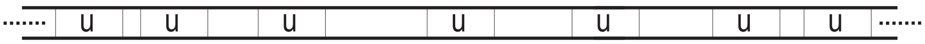
Note that obviously we must have $0 \leq p \leq |u| - m$. We also have the following obvious facts:

- If m-blocking word u is a subword of v then also v is m-blocking.
- If u is m-blocking then it is also k-blocking for all $k \leq m$.

Example 44. In the majority CA (elementary CA number 232) words 11 and 00 are both 2-blocking. □

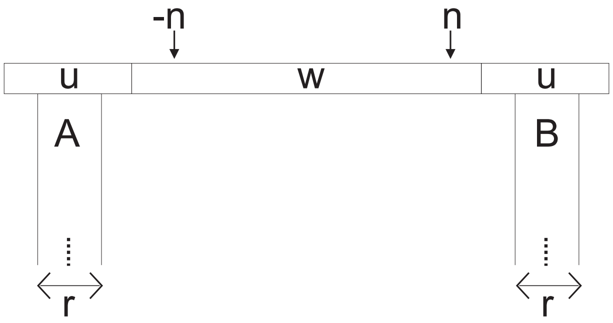
We are mostly interested in r-blocking words where r is a neighborhood radius of the CA. Such words namely prevent the propagation of any change from one side of the blocking word to the other side.

Lemma 87 *Let G be a one-dimensional CA with neighborhood radius r. Suppose G has an r-blocking word u. Let c be any configuration that contains infinitely many copies of u to the right and to the left:*



Such c is equicontinuous.

Proof. Let $n \geq 1$ be arbitrary. Configuration c contains copies of word u to the right of position n and to the left of position $-n$. Let uwu be a subpattern of c where the first and the last u are positioned to the left and to the right of positions $-n$ and n, respectively.



Let e be any configuration with the same subpattern uwu in the same position as c. Because u is r-blocking, there are segments $A = \{a{-}r{+}1, a{-}r{+}2, \dots a\}$ and $B = \{b, b{+}1, \dots b{+}r{-}1\}$ of length r, where $a \leq -n$ and $b \geq n$, such that $G^t(e)(i) = G^t(c)(i)$ for all $t \geq 0$ and all $i \in A$ and $i \in B$. Based on the fact that r is the neighborhood radius of G, we easily see using induction on t that $G^t(e)(i) = G^t(c)(i)$ for all $t \geq 0$ and all $a < i < b$. □

Proposition 88 *Let G be a one-dimensional CA, defined by a radius-r local rule. Then exactly one of the following two alternatives holds:*

(1) $\mathcal{E}$ is residual. This is equivalent to the existence of an r-blocking word.

(2) $\mathcal{E} = \emptyset$. This is equivalent to the sensitivity of G.

Proof. (i) Suppose first that G has an r-blocking word u. Let us prove that $\mathcal{E}$ is residual. For any finite $E \subseteq \mathbb{Z}$ denote by

$$U_E = \{c \in S^{\mathbb{Z}} \mid \exists \text{ finite } D \subseteq \mathbb{Z} \text{ such that } e \in \mathrm{Cyl}(c, D) \Longrightarrow \forall n \geq 1 : G^n(e) \in \mathrm{Cyl}(G^n(c), E)\,\}$$

the set of points that satisfy the equicontinuity condition w.r.t the observation window E. Set U_E is open because if $c \in U_E$ then all configurations of the corresponding cylinder $\mathrm{Cyl}(c, D)$ are also in U_E. Clearly

$$\mathcal{E} = \bigcap_{E \subseteq \mathbb{Z} \text{ finite}} U_E,$$

so it is enough to show that sets U_E are dense. Let $w \in S^*$ be any finite word. By Lemma 87 the configuration $\dots uu\ w\ uu \dots$ is an equicontinuity point, so $\mathcal{E}$ is dense. But $\mathcal{E} \subseteq U_E$ so all U_E are dense as well. We conclude that $\mathcal{E}$ is residual.

(ii) Suppose then that G is not sensitive. Let us show that G has an r-blocking word. Let $E = \{-m, -m+1, \dots, m\}$ where m is such that $2m+1 \geq r$. Due to non-sensitivity, there exists $c \in S^{\mathbb{Z}}$ and domain $D = \{-n, -n+1, \dots, n\}$, $n \geq m$, such that for all $e \in \mathrm{Cyl}(c, D)$ and all $n \geq 1$ holds $G^n(e) \in \mathrm{Cyl}(G^n(c), E)$. But this means that the word $c(-n)c(-n+1)\dots c(n)$ is a $(2m+1)$-blocking word, and hence also an r-blocking word.

Now we have all the ingredients needed to prove the proposition. The four implications in

$$\exists\ r\text{-blocking word} \implies \mathcal{E} \text{ residual} \implies \mathcal{E} \neq \emptyset \implies G \text{ is not sensitive} \implies \exists\ r\text{-blocking word}$$

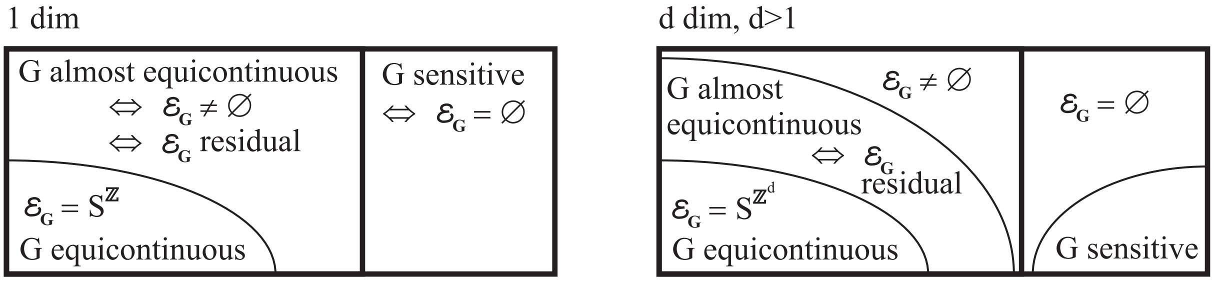
are part (i) above, trivial, Proposition 86, and part (ii) above, respectively. So all four conditions in the implication cycle are equivalent. The proposition now follows from the simple observation that either $\mathcal{E} = \emptyset$ or $\mathcal{E} \neq \emptyset$, but not both. □

If $\mathcal{E}$ is residual then the CA is called *almost equicontinuous*. A one-dimensional CA is either sensitive or almost equicontinuous.

Note: In the two-dimensional case Proposition 88 does not hold. In fact,

- there are two-dimensional CA such that $\mathcal{E} \neq \emptyset$ but $\mathcal{E}$ is not dense,
- there are two-dimensional non-sensitive CA such that $\mathcal{E} = \emptyset$.

We skip these examples. The following picture illustrates the difference in one- and higher dimensional cellular automata:


1 dim
G almost equicontinuous
⇔ $\mathcal{E}_G \neq \emptyset$
⇔ $\mathcal{E}_G$ residual
$\mathcal{E}_G = S^{\mathbb{Z}}$
G equicontinuous
G sensitive
⇔ $\mathcal{E}_G = \emptyset$
d dim, d>1
G almost equicontinuous
$\mathcal{E}_G \neq \emptyset$
⇔ $\mathcal{E}_G$ residual
$\mathcal{E}_G = S^{\mathbb{Z}^d}$
G equicontinuous
$\mathcal{E}_G = \emptyset$
G sensitive

A very strong form of sensitivity is expansivity. A CA G is called *positively expansive* if

$$\exists \varepsilon > 0 : c \neq e \implies \exists n \geq 0 : d(G^n(c), G^n(e)) > \varepsilon.$$

In other words, there exists a finite observation window $E \subseteq \mathbb{Z}^d$ such that for any distinct configurations c and e there exists time $n \geq 0$ such that configurations $G^n(c)$ and $G^n(e)$ differ in some cell in the window E.

If we allow in the definition of positive expansivity the integer n take also negative values we obtain the definition of expansivity. This concept is only defined for reversible CA. More precisely, a reversible CA is called *expansive* if there exists a finite E such that for any two different configurations c and e there exists an integer n (which may be also negative) such that $G^n(c)$ and $G^n(e)$ differ in some cell in domain E.

Example 45. Non-zero translations are not positively expansive but in the one-dimensional case they are expansive. The *xor* CA is not positively expansive because differences only propagate to the left, but the three neighbor *xor* (elementary CA 150, local rule $f(a, b, c) \equiv a + b + c \pmod 2$) is positively expansive. $\square$

Our first observation is that expansivity and positive expansivity are stronger conditions than sensitivity to initial conditions.

Proposition 89 *Expansive and positively expansive CA are sensitive.*

Proof. For positive expansivity this is clear since the space $S^{\mathbb{Z}^d}$ does not have any isolated points. If E is the finite observation window confirming positive expansivity of G then the same E also confirms sensitivity: For any finite $D \subseteq \mathbb{Z}^d$ and any $c \in S^{\mathbb{Z}^d}$ there exist a configuration $e \in \mathrm{Cyl}(c, D)$ such that $e \neq c$. By positive expansivity, $G^n(e)(\vec{x}) \neq G^n(c)(\vec{x})$ for some $n \geq 0$ and $\vec{x} \in E$. This proves sensitivity.

Suppose then that reversible G is expansive, and let E be a finite observation window confirming its expansivity. Let $D \subseteq \mathbb{Z}^d$ and $c \in S^{\mathbb{Z}^d}$ be arbitrary. It is enough to show that for some $e \in \mathrm{Cyl}(c, D)$, some $n \geq 0$ and some $\vec{x} \in E$ we have $G^n(e)(\vec{x}) \neq G^n(c)(\vec{x})$. By expansivity, such n and $\vec{x}$ exist for every $e \neq c$, except that time n may be negative. If $n \geq 0$ the proof is complete, so suppose now that $G^n(e)(\vec{x}) \neq G^n(c)(\vec{x})$ where $n < 0$. In reversible CA periodic points are dense, so there exist periodic c' and e' such that $c', e' \in \mathrm{Cyl}(c, D)$, and $G^n(e')(\vec{x}) = G^n(e)(\vec{x}) \neq G^n(c)(\vec{x}) = G^n(c')(\vec{x})$. If p is a common period for c' and e' we have that $G^{n+kp}(e')(\vec{x}) = G^n(e')(\vec{x}) \neq G^n(c')(\vec{x}) = G^{n+kp}(c')(\vec{x})$, for every $k \in \mathbb{Z}$. For sufficiently large k the time $n + kp$ is positive, so either c' or e' confirms the sensitivity. $\square$

It is also easy to see that positively expansive CA must be surjective.

Proposition 90 *Positively expansive CA G is surjective.*

Proof. By the Garden-of-Eden -theorem, if G is not surjective then there are two configurations c, e such that $G(c) = G(e)$ but the difference set

$$\mathit{diff}(c, e) = \{\vec{n} \in \mathbb{Z}^d \mid c(\vec{n}) \neq e(\vec{n})\ \}$$

is finite and non-empty. By translation invariance, for any observation window E we can choose c and e in such a way that $E \cap \mathit{diff}(c, e) = \emptyset$. This contradicts positive expansivity, as the difference of c and e is never seen inside the window E. $\square$

The following two propositions show that only one-dimensional CA can be positively expansive or expansive, and that reversible CA can never be positively expansive.

Proposition 91 *For $d \geq 2$, no d-dimensional CA is (positively) expansive.*

Proof. We only prove this for positive expansivity. The proof for expansivity is similar.

Suppose that G is a positively expansive d-dimensional CA, where $d \geq 2$. Let $E \subseteq \mathbb{Z}^d$ be a finite observation window that confirms the positive expansivity. As we may replace E by any of its supersets, we may assume that E is a size $(2k+1)^d$ hypercube, centered at the origin.

For every $i = 1, 2, \ldots$, let D_i denote the size $(2i+1)^d$ hypercube centered at the origin. Let us use the compactness of $S^{\mathbb{Z}^d}$ to show that for some $n \in \mathbb{Z}_+$ we have the following property: If c and e are any two configurations such that

$$G^t(c)(\vec{x}) = G^t(e)(\vec{x}) \text{ for all } 0 \leq t \leq n \text{ and all } \vec{x} \in D_k$$

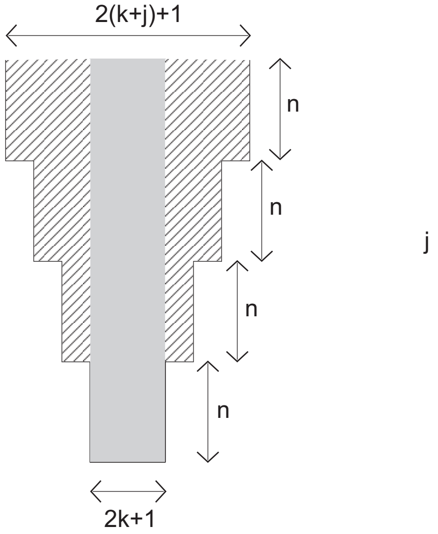
then

$$c(\vec{y}) = e(\vec{y}) \text{ for all } \vec{y} \in D_{k+1}.$$

Suppose the contrary: for every n there exist $c_n, e_n \in S^{\mathbb{Z}^d}$ such that $G^t(c_n)(\vec{x}) = G^t(e_n)(\vec{x})$ for all $0 \leq t \leq n$ and all $\vec{x} \in D_k$ but $c_n(\vec{y}) \neq e_n(\vec{y})$ for some $\vec{y} \in D_{k+1}$. Let c, e be the limit of a converging subsequence of pairs c_n, e_n, for $n = 1, 2, \ldots$. Then $c(\vec{y}) \neq e(\vec{y})$ for some $\vec{y} \in D_{k+1}$ but $G^t(c)(\vec{x}) = G^t(e)(\vec{x})$ for all $t \geq 0$ and all $\vec{x} \in D_k = E$. This contradicts the positive expansivity of G.

So we have number n such that states $G^t(c)(\vec{x})$ for $0 \leq t \leq n$ and $\vec{x} \in D_k$ uniquely determine $c(\vec{y})$ for all $\vec{y} \in D_{k+1}$. But then we also have the property that for any $j \geq 0$ the states $G^t(c)(\vec{x})$ for $0 \leq t \leq n$ and $\vec{x} \in D_{k+j}$ uniquely determine $c(\vec{y})$ for $\vec{y} \in D_{k+j+1}$. This is because every cell $\vec{y} \in D_{j+k+1}$ belongs to a size $(2k+3)^d$ hypercube whose co-centric hypercube of size $(2k+1)^d$ is inside D_{k+j}.

So in the space-time diagram



the states in the gray area uniquely determine the shaded area. In other words, for every $j \geq 0$ the following holds: if $c(\vec{y}) \neq e(\vec{y})$ for some $\vec{y} \in D_{k+j}$ then $G^t(c)(\vec{x}) \neq G^t(e)(\vec{x})$ for some $0 \leq t \leq jn$ and some $\vec{x} \in D_k$.

A contradiction now arises from the fact that if $d \geq 2$ the volume of D_{k+j} grows faster than the size of $D_k \times \{0, 1, \dots, jn\}$ when j increases. More precisely, for some sufficiently large j,

$$|D_{k+j}| = (2k + 2j + 1)^d > (2k + 1)^d (jn + 1) = |D_k \times \{0, 1, \dots, jn\}|.$$

By the pigeon-hole principle then there exists configurations c, e that are not identical inside D_{k+j} but $G^t(c)(\vec{x}) = G^t(e)(\vec{x})$ for all $0 \leq t \leq jn$ and all $\vec{x} \in D_k$, a contradiction. □

Proposition 92 *A reversible CA is not positively expansive.*

Proof. By Proposition 91 it is enough to consider a one-dimensional reversible CA G. Suppose that G would be positively expansive, and let $E = \{-m, -m + 1, \dots, m\}$ be an observation window confirming positive expansivity. Let r be the neighborhood radius of its inverse G^{-1}.

By compactness (as in the proof of Proposition 91) we see that for some $n \geq 1$ we have that if c and e are any two configurations such that

$$G^t(c)(i) = G^t(e)(i) \text{ for all } 0 \leq t \leq n \text{ and all } -m \leq i \leq m$$

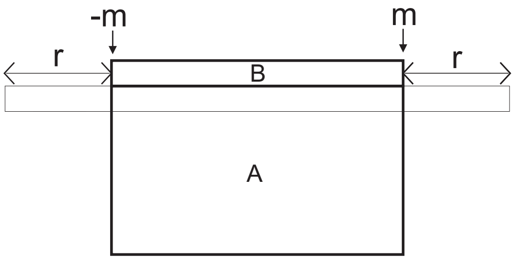
then

$$c(i) = e(i) \text{ for all } -m - r \leq i \leq m + r.$$

But because r is the neighborhood radius of G^{-1}, this in turn implies that

$$G^{-1}(c)(i) = G^{-1}(e)(i) \text{ for all } -m \leq i \leq m.$$

In other words, in the space-time diagram

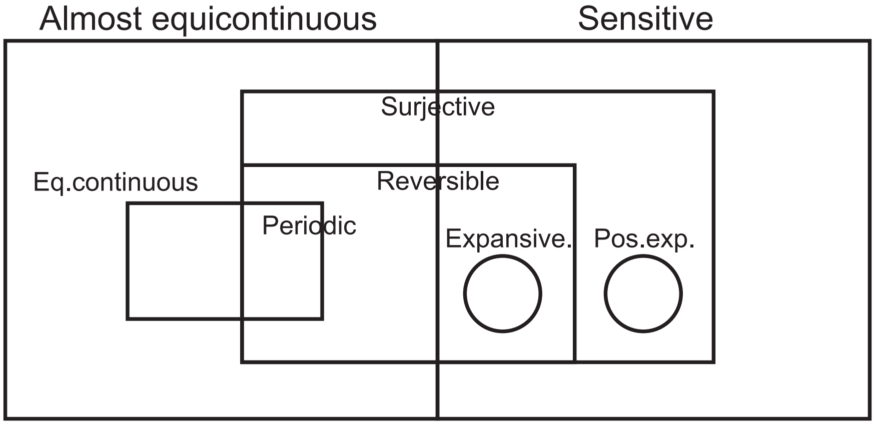


states in region A uniquely determine the states in region B. Inductively we can conclude that

$$G^t(c)(i) = G^t(e)(i) \text{ for all } t < 0 \text{ and } -m \leq i \leq m,$$

so all states above segment B are also uniquely determined. This implies that G^{-1} is eventually periodic and hence, due to its reversibility, periodic. Clearly, a periodic CA cannot be positively expansive. □

The following diagram summarizes the sensitivity results we obtained in this section for one-dimensional CA:



In higher dimensions the circles representing expansive and positively expansive CA are missing, and the complement of sensitive CA is not the set of almost equicontinuous CA. Other aspects of the diagram are valid in higher dimensions as well.

5.6 Mixing properties

Mixing of the configuration space is another property associated to chaos. As sensitivity, also mixing comes in different variants. Cellular automaton G is called *transitive* if for all non-empty open sets U and V there exists $n \geq 0$ such that $G^n(U) \cap V \neq \emptyset$. As cylinders form a basis, it is sufficient to check this condition for all non-empty cylinders U and V. If G is non-surjective then its transitivity is contradicted by non-empty open sets U and V such that $V \cap G(S^{\mathbb{Z}^d}) = \emptyset$ and $U \cap V = \emptyset$. So only surjective CA can be transitive.

Note that the condition $G^n(U) \cap V \neq \emptyset$ can be replaced by the equivalent condition $U \cap G^{-n}(V) \neq \emptyset$. Also, we can require n to be strictly positive since we can use $G^{-1}(V)$ in place of V.

Example 46. The *xor* -cellular automaton is easily seen transitive. For any fixed word w of length n, define the mapping $h : S^n \longrightarrow S^n$ so that $u \mapsto v$ iff $G^n(\ldots w\ u \ldots) = \ldots v \ldots$ where v is a word of length n starting in the same position as w, only n time steps later. Mapping h is injective, and therefore also surjective. We see that for any two cylinders U and V with domain $\{1, 2, \ldots, n\}$ holds $G^n(U) \cap V \neq \emptyset$. This proves transitivity. □

Transitivity is a stronger property that sensitivity:

Proposition 93 *A transitive CA (with at least two states) is sensitive to initial conditions.*

Proof. Let G be a CA that is not sensitive. Because G is not sensitive there exists some configuration $c \in S^{\mathbb{Z}^d}$ and a finite domain $D \subseteq \mathbb{Z}^d$ such that

$$e \in \mathrm{Cyl}(c, D) \implies G^t(e)(\vec{0}) = G^t(c)(\vec{0}) \text{ for all } t \geq 0.$$

Denote $C = \mathrm{Cyl}(c, D)$. Let $\tau = \tau_{\vec{n}}$ be a translation by some vector $\vec{n} \neq \vec{0}$ such that $U = \tau^{-1}(C) \cap C \neq \emptyset$. Such a translation exists because the domain D is finite. Set U is a non-empty cylinder. Let $e \in U$ be arbitrary, so $e \in C$ and $\tau(e) \in C$. For all $t \geq 0$

$$G^t(e)(\vec{n}) = \tau(G^t(e))(\vec{0}) = G^t(\tau(e))(\vec{0}) = G^t(c)(\vec{0}) = G^t(e)(\vec{0}).$$

We conclude that $G^t(U) \cap V = \emptyset$ for all $t \geq 0$ where V is any cylinder with domain $\{\vec{0}, \vec{n}\}$ and with different state assigned to $\vec{0}$ and $\vec{n}$. Hence G is not transitive. □

Example 47. As an example of a surjective CA that is sensitive but not transitive, consider the product $\tau \times I$ of a non-zero translation τ and the identity function I, both over the binary state set $\{0, 1\}$. This CA is sensitive because τ is sensitive, but it is not transitive because I is not transitive. □

The following proposition characterizes transitive systems in terms of orbits. Let us denote by

$$\mathcal{T}_G = \mathcal{T} = \{c \in S^{\mathbb{Z}^d} \mid \text{ the orbit } c, G(c), G^2(c), \ldots \text{ is dense}\}$$

the set of *transitive points* of G, i.e., those points whose orbit is dense.

Proposition 94 *The following three conditions are equivalent for a CA G:*

(i) G is transitive,

(ii) $\mathcal{T}_G$ is a residual set,

(iii) $\mathcal{T}_G \neq \emptyset$.

Proof. "(i) $\implies$ (ii)" Suppose that G is transitive. Let $U_1, U_2, \ldots$ be the set of all cylinders. For each cylinder U_n, let

$$X_n = \{c \in S^{\mathbb{Z}^d} \mid G^m(c) \in U_n \text{ for some } m \geq 0 \ \}$$

be the set of all points whose forward orbits intersect U_n. Then

$$\mathcal{T}_G = \cap_{i=0}^{\infty} X_n,$$

so it is sufficient so show that each X_n is open and dense. If $c \in X_n$ then there is m such that $G^m(c) \in U_n$. Function G^m is continuous, so $G^{-m}(U_n)$ is an open neighborhood of c that is a subset of X_n. Hence X_n is open. For denseness consider an arbitrary open set U. By transitivity of G there exists m such that $G^m(U) \cap U_n \neq \emptyset$, that is, $U \cap X_n \neq \emptyset$.

"(ii) $\Longrightarrow$ (iii)" follows form the fact that $S^{\mathbb{Z}^d}$ is a Baire space.

"(iii) $\Longrightarrow$ (i)" Suppose that c has dense forward orbit, and let U and V be arbitrary non-empty open sets. By the denseness of the orbit we have that $G^n(c) \in U$ for some n. Because $V' = V \setminus \{c, G(c), G^2(c), \ldots, G^n(c)\}$ is non-empty and open, there exists m such that $G^m(c) \in V'$. Clearly $m > n$ and $G^m(c) \in V$, so $G^{m-n}(U) \cap V \neq \emptyset$. Hence the CA is transitive. □

We can make the following easy observation concerning attractors of transitive CA:

Proposition 95 *The limit set $S^{\mathbb{Z}^d}$ is the only attractor of a transitive CA.*

Proof. Suppose U is a clopen set satisfying $G(U) \subseteq U$. If U is not the whole space $S^{\mathbb{Z}^d}$ then there is another clopen set V such that $U \cap V = \emptyset$. But since $G^n(U) \subseteq U$ for all n, we have $G^n(U) \cap V = \emptyset$ for all n, which means that G is not transitive. We conclude that a transitive CA can only have one attractor, specified by $U = S^{\mathbb{Z}^d}$. □

The following examples shows that the converse is not true.

Example 48. Consider the CA $G = H \circ \sigma$ due to Coven and Hedlund, where H is the elementary CA number 180 and σ is the left shift. It is a one-dimensional, binary state CA with neighborhood $(0, 1, 2)$. In H, the state of a cell is swapped if and only if it is followed by word 10.

Let us show that H is surjective and non-sensitive (and hence non-transitive) but the limit set $\Omega = S^{\mathbb{Z}}$ is the only attractor. The surjectivity of H is clear since the elementary CA 180 is left-permutive. Non-sensitivity follows from the fact that 000 is a two-blocking word (exercise, based on the fact that the only preimage of word $w_n = 00(11)^n10$ that begins with 00 is $w_{n+1} = 00(11)^{n+1}10$.)

Let $E = \{a, a+1, \ldots, b\}$ be any finite contiguous domain, and let $C_1, C_2, \ldots, C_n$ be the cylinders with domain E. If we build a directed graph with n vertices $C_1, C_2, \ldots, C_n$, and put an edge $C_i \longrightarrow C_j$ if and only if $G(C_i) \cap C_j \neq \emptyset$ then one can show that the graph is strongly connected. We say that the CA is *chain transitive.* If $U \neq \emptyset, S^{\mathbb{Z}}$ is any clopen set then it is a union of some cylinders C_i for some domain E. Due to chain transitivity $G(C_i) \cap C_j \neq \emptyset$ for some $C_i \subseteq U$ and $C_j \subseteq S^{\mathbb{Z}} \setminus U$. Hence $G(U) \not\subseteq U$. It follows that the limit set $S^{\mathbb{Z}}$ is the only attractor. □

Motivated by the example above, let us define a weaker form of transitivity. A CA is called *chain transitive* if for every $c, e \in S^{\mathbb{Z}^d}$ and every finite $D \subseteq \mathbb{Z}^d$ there exists a sequence $c_0, c_1, \ldots, c_n$ of configurations such that $c_0 = c$, $c_n = e$ and for every $i = 1, 2, \ldots, n$ we have $G(c_{i-1}) \in \mathrm{Cyl}(c_i, D)$. In other words, for all domains D, the following directed graph must be strongly connected: The vertices are the cylinders C_i with domain D, and there is an edge $C_i \longrightarrow C_j$ iff $G(C_i) \cap C_j \neq \emptyset$.

Chain transitive CA are surjective as they can have no orphans. Clearly every transitive CA is chain transitive.

Proposition 96 *A CA G is chain transitive if and only if the only $S^{\mathbb{Z}^d}$ is the only attractor of G.*

Proof. If G is not chain transitive then for some domain D the directed graph constructed for that domain is not strongly connected. Let C_i and C_j be vertices such that there is no path in the graph from C_i to C_j. Let U be the union of all the cylinders that can be reached in the graph from C_i. Then U is clopen as a finite union of cylinders, $U \neq S^{\mathbb{Z}^d}$ because $A \cap C_j = \emptyset$, and $G(U) \subseteq U$. Hence U specifies an attractor different from $S^{\mathbb{Z}^d}$.

Conversely, suppose that G is chain transitive. Let U be a clopen set such that $G(U) \subseteq U$, and suppose that $U \neq \emptyset$ and $U \neq S^{\mathbb{Z}}$. Let $D \subseteq \mathbb{Z}^d$ be a finite domain such that U is a disjoint union of cylinders with domain D. By chain transitivity there are cylinders C_i and C_j with domain D such that $C_i \subseteq U$, $C_j \subseteq S^{\mathbb{Z}} \setminus U$ and $G(C_i) \cap C_j \neq \emptyset$. Hence $G(U) \not\subseteq U$, a contradiction. We conclude that $U = S^{\mathbb{Z}^d}$ is the only non-empty clopen set satisfying $G(U) \subseteq U$, so $S^{\mathbb{Z}^d}$ is the only attractor. □

Mixing is a more restrictive concept than transitivity: Cellular automaton G is called *mixing* if for all non-empty open sets U and V there exists positive integer n such that $G^k(U) \cap V \neq \emptyset$ for all $k \geq n$. Again, it is sufficient to verify this condition in the case U and V are cylinders. It is immediate from the definition that every mixing CA is also transitive:

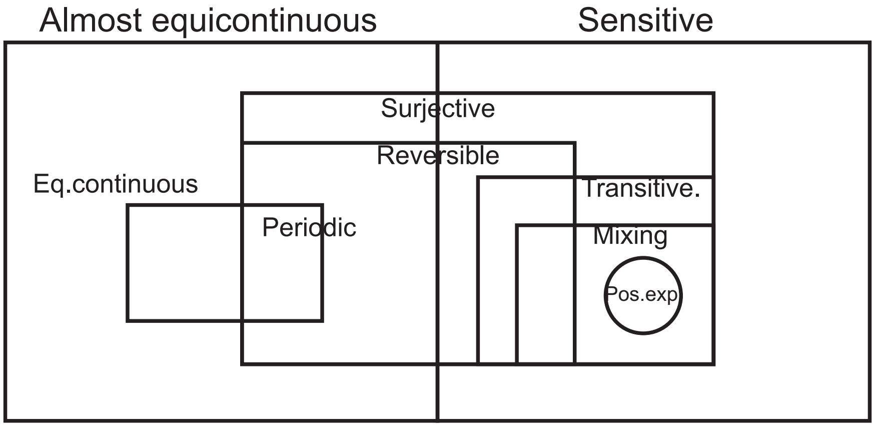
Proposition 97 *A mixing CA is transitive.* □

Without a proof, we state the following implication. For a proof, see e.g.

P.Kůrka. *Topological and Symbolic Dynamics.* Societe Mathematique de France, 2003.

Proposition 98 *A positively expansive CA is mixing.* □

We can now add transitivity and mixing to the sensitivity classification diagram:



The relationship of reversible, expansive CA to transitivity and mixing is not known, so expansive CA were excluded from the diagram.

Note that the concepts concerning sensitivity and transitivity are not just CA related notions, but they are concepts defined in the same way in the context of general topological dynamical systems. (The sensitivity related concepts require that the space has a metric, while for transitivity and mixing only the topology is needed.) Devaney defines a chaotic dynamical system to be a system that

1. is sensitive to initial conditions,

2. is transitive, and

3. temporally periodic points are dense.

We proved in proposition 93 that transitivity always implies sensitivity in cellular automata, so transitivity and dense periodic orbits characterize chaotic CA. It is also conjectured (but not proven!) that all surjective CA have dense periodic orbits, and if true, this would imply that in cellular automata transitivity is equivalent to chaos.

Open Problem. Is it true that in every surjective CA every cylinder contains a temporally periodic configuration ? □

Among reversible CA temporally periodic configurations are dense (because spatially totally periodic configurations are also temporally periodic), so among reversible CA Devaney's chaos is equivalent to transitivity.

5.7 Measure invariance in surjective CA

One useful tool not yet used by us is the invariance of the uniform measure under surjective CA (i.e., the balance property stated in Proposition 13). Let us start by recalling some basic definitions of ergodic theory. Let us denote $X = S^{\mathbb{Z}^d}$. A non-empty family Σ of subsets of X is called a *σ-algebra* if its is closed under complementation and countable unions, that is, if

$$\begin{array}{rcl} A \in \Sigma & \Longrightarrow & X \setminus A \in \Sigma \\ A_1, A_2, \ldots \in \Sigma & \Longrightarrow & A_1 \cup A_2 \cup \ldots \in \Sigma. \end{array}$$

The *Borel σ-algebra* $\mathcal{B}$ is the smallest σ-algebra that contains all open sets. (It is well defined as the intersection of all σ-algebras that contain the open sets.) From now on, $\mathcal{B}$ denotes the Borel σ-algebra, and its elements are called Borel sets or (Borel) measurable sets. The following results (Dynkin's π-λ Theorem) is needed later. For a proof, see any textbook on measures:

Lemma 99 *Let Γ be a family of subsets of X that is closed under finite intersections (so-called π-system). Let Δ be a family of subsets of X that contains X and is closed under complementation and countable unions of pairwise disjoint sets (so-called λ-system). If $\Gamma \subseteq \Delta$ then there exists a σ-algebra Σ such that $\Gamma \subseteq \Sigma \subseteq \Delta$.* □

For any continuous function $G : X \longrightarrow X$ and any Borel measurable set A, the set $G^{-1}(A)$ is also Borel measurable. (The family Σ consisting of all $A \subseteq X$ such that $G^{-1}(A) \in \mathcal{B}$ is easily seen to be a σ-algebra. For open U the set $G^{-1}(U)$ is open and hence in $\mathcal{B}$, so all open sets are in Σ. It follows that $\mathcal{B} \subseteq \Sigma$.) In particular, pre-images of measurable sets under CA functions are measurable.

A *Borel probability measure* is a function $\mu : \mathcal{B} \longrightarrow \mathbb{R}$ that satisfies the following conditions:

- $\mu(\emptyset) = 0$, $\mu(X) = 1$, and $\mu(A) \geq 0$ for all $A \in \mathcal{B}$, and
- (σ-additivity) If $A_1, A_2, \ldots$ are pairwise disjoint elements of $\mathcal{B}$ then

$$\mu(A_1 \cup A_2 \cup \ldots) = \mu(A_1) + \mu(A_2) + \ldots$$

Note that σ-additivity applies to countable families of pairwise disjoint Borel sets.

Example 49. Let $c \in X$ be fixed, and define

$$\mu(A) = \begin{cases} 1, & \text{if } c \in A, \\ 0, & \text{if } c \notin A. \end{cases}$$

This is easily seen to be a measure, the point measure. More generally, if $C \subseteq X$ is finite we can define a measure by

$$\mu(A) = \frac{|A \cap C|}{|C|}.$$

□

Let us first show that the values of a measure on cylinders uniquely determines the measure:

Proposition 100 *If two Borel probability measures agree on cylinders then they are identical.*

Proof. Let Γ be the set of cylinders. Then Γ is closed under finite intersections. Suppose μ_1 and μ_2 be two Borel probability measures such that $\mu_1(C) = \mu_2(C)$ for all $C \in \Gamma$. Let Δ be the family of Borel sets A such that $\mu_1(A) = \mu_2(A)$. All cylinders are in Δ. If $A \in \Delta$ then

$$\mu_1(X \setminus A) = 1 - \mu_1(A) = 1 - \mu_2(A) = \mu_2(X \setminus A),$$

so the complement of A is in Δ. If $A_1, A_2, \ldots$ are pairwise disjoint elements of Δ then

$$\mu_1(A_1 \cup A_2 \cup \ldots) = \mu_1(A_1) + \mu_1(A_2) + \ldots = \mu_2(A_1) + \mu_2(A_2) + \ldots = \mu_2(A_1 \cup A_2 \cup \ldots).$$

We see that Δ and Γ satisfy the conditions in Lemma 99, so there exists a σ-algebra Σ such that $\Gamma \subseteq \Sigma \subseteq \Delta$. As all open sets are countable unions of cylinders we see that Σ contains all open sets, and therefore all Borel sets. We conclude that $\mu_1(A) = \mu_2(A)$ for all Borel sets A. □

For any Borel probability measure μ and any continuous $G : X \longrightarrow X$ we define a new measure $G(\mu)$ as follows:

$$G(\mu)(A) = \mu(G^{-1}(A)) \text{ for all } A \in \mathcal{B}.$$

The function $G(\mu)$ is defined on every Borel set A because $G^{-1}(A)$ is Borel. So defined $G(\mu)$ is a Borel probability measure:

Lemma 101 *For any Borel probability measure μ on X and any continuous $G : X \longrightarrow X$ the function $G(\mu)$ is a Borel probability measure on X.*

Proof. Follows directly from the facts that $G^{-1}(\emptyset) = \emptyset$, $G^{-1}(X) = X$, $G^{-1}(X \setminus A) = X \setminus G^{-1}(A)$ and

$$G^{-1}(A_1 \cup A_2 \cup \ldots) = G^{-1}(A_1) \cup G^{-1}(A_1) \cup \ldots.$$

Moreover, we need to note that $G^{-1}(A_i)$ and $G^{-1}(A_j)$ are disjoint if A_i and A_j are disjoint. □

The measure $G(\mu)$ describes the probability distribution of configurations after one application of G, if the configurations were originally drawn according to distribution μ: The $G(\mu)$-probability that a configuration belongs to a measurable set A after one application of G is the same as the μ-probability that it originally belonged to $G^{-1}(A)$.

We have seen that any CA G defines a function $\mu \mapsto G(\mu)$ on Borel probability measures. It turns out that this function can be viewed as a dynamical system because the Borel probability measures can be given a compact metric such this application is continuous. We do not go into any further details of this interesting aspect.

We say that measure μ is *invariant* under G if $G(\mu) = \mu$. Ergodic theory is a field that studies dynamics under invariant measures. In the following we show that all surjective CA have a very simple invariant measure, the uniform measure.

The simplest Borel probability measures are the *Bernoulli* probability measures. Given a probability distribution $p : S \longrightarrow [0, 1]$ that satisfies $\sum_{a \in S} p_a = 1$, the corresponding Bernoulli measure μ_p gives cylinder $C = \mathrm{Cyl}(c, D)$ the value

$$\mu_p(C) = \prod_{\vec{n} \in D} p(c(\vec{n})).$$

We skip the proof that this assignment on cylinders extends into a probability measure on Borel sets. (See any book on measures.)

A particular case of Bernoulli measures is the *uniform* probability measure that uses $p(a) = \frac{1}{|S|}$ for all $a \in S$. In this case

$$\mu_p(C) = \left(\frac{1}{|S|}\right)^{|D|},$$

so each cylinder with the same domain is given the same probability. Under this probability measure "each configuration has the same probability".

Proposition 102 *Let G be a surjective CA and let μ be the uniform Bernoulli measure. Then $G(\mu) = \mu$.*

Proof. This is another way of stating the balance property of Proposition 13. From Proposition 100 we know that it is enough to show that $G(\mu)(C) = \mu(C)$ for all cylinders $C = \mathrm{Cyl}(c, D')$ with arbitrary finite domain D'. By Proposition 13

$$G^{-1}(C) = C_1 \cup C_2 \cup \ldots C_n$$

where C_i are disjoint cylinders with some domain D, and $n = |S|^{|D|-|D'|}$. We have

$$\mu(G^{-1}(C)) = \mu(C_1) + \mu(C_2) + \ldots + \mu(C_n) = n \left(\frac{1}{|S|}\right)^{|D|} = \left(\frac{1}{|S|}\right)^{|D'|} = \mu(C).$$

□

As the first application of the measure invariance we prove the following fact:

Proposition 103 *Attractors of surjective CA G are exactly the clopen sets U that satisfy $G(U) = U$. The complement of an attractor is then also an attractor (or the empty set).*

Proof. Let G be a surjective CA, and let U be any clopen set such that $G(U) \subseteq U$. Then $U \subseteq G^{-1}(U)$. Let us prove that $G^{-1}(U) = U$. Denote $V = G^{-1}(U) \setminus U$. By Proposition 102

$$\mu(V) = \mu(G^{-1}(U)) - \mu(U) = 0,$$

for the unform measure μ. But V is clopen, and the only open set of measure 0 is the empty set. We conclude that $G^{-1}(U) = U$. By surjectivity, $G(U) = U$ and $G(S^{\mathbb{Z}^d} \setminus U) = S^{\mathbb{Z}^d} \setminus U$.
□

It also follows now that the full space $S^{\mathbb{Z}^d}$ is the only *subshift attractor* (i.e. a translation invariant attractor) of a surjective CA.

Corollary 104 *The only translation invariant attractor of a surjective CA is the full space $S^{\mathbb{Z}^d}$.*

Proof. By Proposition 103, all attractors of a surjective CA are clopen. But the only translation invariant clopen sets are $S^{\mathbb{Z}^d}$ and the empty set. To see this, let C be a translation invariant clopen set, and suppose that $C \neq S^{\mathbb{Z}^d}$. Let $U \neq \emptyset$ be an open set such that $U \cap C = \emptyset$. For any translation τ, we have $\tau(U) \cap C = \tau(U) \cap \tau(C) = \tau(U \cap C) = \emptyset$. But then $C = \emptyset$ because for any non-empty open sets A and B there exists a translation τ such that $\tau(A) \cap B \neq \emptyset$. □

One of the basic theorems of Ergodic theory is the *Poincaré recurrence theorem.* It deals with measure invariant transformations so it directly applies to surjective CA. We first prove a variant which states that among the points inside any measurable set A, the set of points that do not return back to A infinitely many times has measure zero:

Proposition 105 *Let G be a surjective CA and let μ denote the uniform measure. For every Borel set A the set*

$$B = \{c \in A \mid \exists k: \ \forall n > k : G^n(c) \notin A\}$$

of points of A that only return finitely many times to A has measure $\mu(B) = 0$.

Proof. For every $k \geq 0$ denote by

$$A_k = G^{-k}(A) \cup G^{-k-1}(A) \cup G^{-k-2}(A) \cup \dots$$

the set of points that visit A after at least k applications of G. These sets form a decreasing chain

$$A_0 \supseteq A_1 \supseteq A_2 \supseteq \dots$$

under inclusion. Each A_k is a Borel set as a countable union of Borel sets $G^{-i}(A)$. For every k we have $G^{-1}(A_k) = A_{k+1}$. Because the uniform measure is invariant, we have $\mu(A_k) = \mu(A_{k+1})$. This means that all A_k have the same measure $\mu(A_k) = \mu(A_0)$.

Because $A \subseteq A_0$, we have that $A \setminus A_k \subseteq A_0 \setminus A_k$ for every k, and consequently

$$\mu(A \setminus A_k) \leq \mu(A_0 \setminus A_k) = \mu(A_0) - \mu(A_k) = 0.$$

Set B consists of those points of of A that do not belong to some A_k, that is,

$$B = \bigcup_{k=0}^{\infty} (A \setminus A_k).$$

But then

$$\mu(B) \leq \sum_{k=0}^{\infty} \mu(A \setminus A_k) = \sum_{k=0}^{\infty} 0 = 0,$$

which completes the proof. □

Remark: In particular, the set of points of A that never return to A has measure zero. (This is the set $A \setminus A_1$ in the previous proof.)

A configuration is called *recurrent* if it returns to each of its open neighborhoods. Denote by

$$\mathcal{R}_G = \mathcal{R} = \{c \in S^{\mathbb{Z}^d} \mid \forall \text{ open } U\text{:} \ \ c \in U \Longrightarrow \exists n \geq 1 : G^n(c) \in U\}.$$

the set of recurrent points. If c is recurrent then it, in fact, returns infinitely many times to all its open neighborhoods U: Either c is periodic, or for every n the set $U\setminus\{G(c), G^2(c), \dots, G^n(c)\}$ is an open neighborhood of c, which then must be visited by c. Another variant of Poincare's recurrence theorem tells us that in surjective CA recurrent points are dense:

Proposition 106 *If G is surjective then $\mathcal{R}$ is a countable intersection of open sets of uniform measure 1. In particular, $\mathcal{R}$ is residual and $\mu(\mathcal{R}) = 1$.*

Proof. Let $U_1, U_2, \dots$ be an enumeration of all cylinders. For each $k = 1, 2, \dots$ define

$$X_k = \{c \in U_k \mid \forall n \geq 1 : G^n(c) \notin U_k\}.$$

By Proposition 105 (or more precisely, the remark following the proposition) we know that $\mu(X_k) = 0$. Set X_k is closed because

$$X_k = U_k \setminus \bigcup_{n=1}^{\infty} G^{-n}(U_k)$$

where U_k is clopen.

Let us prove that

$$X_1 \cup X_2 \cup \dots = S^{\mathbb{Z}^d} \setminus \mathcal{R}.$$

Indeed, if $c \in X_k$ then c cannot be recurrent as U_k is an open neighborhood it never returns to. Conversely, if c is not recurrent then it has an open neighborhood U such that $G^n(c) \notin U$ for all $n = 1, 2, \dots$. As cylinders form a basis, there exists k such that $c \in U_k \subseteq U$. But then $c \in X_k$.

We conclude that

$$\mathcal{R} = \bigcap_{k=1}^{\infty} (S^{\mathbb{Z}^d} \setminus X_k)$$

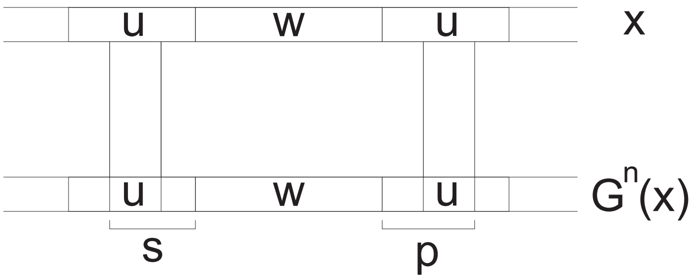
is a countable intersection of open sets with measure $\mu(S^{\mathbb{Z}^d} \setminus X_k) = 1 - \mu(X_k) = 1$.

To complete the proof we note that if $\mu(A) = 1$ then A is dense (because otherwise its complement would contain a cylinder and all cylinders have positive measure). Also any countable intersection of measure 1 sets has measure 1. (Because its complement is a countable union of sets of measure 0.) □

If a surjective CA is non-sensitive, we can now conclude that periodic points are dense:

Proposition 107 *If a one-dimensional CA G is surjective and not sensitive then the periodic configurations are dense in $S^{\mathbb{Z}}$.*

Proof. By proposition 88 there is an r-blocking word u where r is the neighborhood radius of G. Let $w \in S^*$ be an arbitrary word. It is enough to show that there is a periodic configuration that contains pattern w. Let C be a cylinder determined by word uwu. By Proposition 106, cylinder C contains a recurrent configuration x so, in particular, $G^n(x) \in C$ for some $n \geq 1$.



Let $c = \dots wuwuwu \dots$ be the spatially periodic configuration with period wu, contained in cylinder C. Because u is an r-blocking word, configuration $G^n(c)$ contains a pattern swp where p and s are a suffix and prefix of u such that $|p|+|s| \geq |u|$. Because $G^n(c)$ is spatially periodic with period length $|wu|$ we see that $G^n(c) = \dots wuwuwu \dots$. Hence c is temporally periodic. □

A measure theoretic concept that corresponds to transitivity in topological dynamics is ergodicity. We say that CA G is *ergodic* with respect to the uniform measure μ if for Borel sets A holds

$$G^{-1}(A) = A \Longrightarrow \mu(A) = 0 \text{ or } \mu(A) = 1.$$

Recall that we denote by $\mathcal{T}$ the set of transitive points, i.e., the points whose forward orbit is dense. (Topologically) transitive CA were characterized in Proposition 94 as those having transitive points, and it was shown that in this case $\mathcal{T}$ is, in fact, residual. In ergodic CA the set $\mathcal{T}$ of transitive points has measure one:

Proposition 108 *If a CA G ergodic (with respect to the uniform measure μ) then $\mu(\mathcal{T}) = 1$.*

Proof. Let $U_1, U_2, \dots$ be an enumeration of all cylinders. For every k, let

$$X_k = \bigcap_{n=1}^{\infty} \cup_{i=n}^{\infty} G^{-i}(U_k)$$

be the set of points that visit U_k infinitely many times. Sets X_k are Borel sets and satisfy $G^{-1}(X_k) = X_k$. Due to ergodicity, either $\mu(X_k) = 0$ or $\mu(X_k) = 1$.

By the Poincaré recurrence theorem (Proposition 105), $\mu(U_k \setminus X_k) = 0$. Consequently,

$$\mu(X_k) \geq \mu(U_k \cap X_k) = \mu(U_k) - \mu(U_k \setminus X_k) = \mu(U_k) > 0,$$

so $\mu(X_k) = 1$.

The result now follows from the fact that

$$\mathcal{T} = X_1 \cap X_2 \cap \dots.$$

□

A measure one set is always dense, so Propositions 94 and 108 give the following:

Corollary 109 *Ergodic CA are transitive.* □