Fundamenta Informaticae 187(1) : 31–59 (2022)
Available at IOS Press through:
https://doi.org/10.3233/FI-222129

Number Conservation via Particle Flow in One-dimensional Cellular Automata

Markus Redeker*
Hamburg, Germany
markus2.redeker@mail.de

Abstract. A number-conserving cellular automaton is a simplified model for a system of interacting particles. This paper contains two related constructions by which one can find all one-dimensional number-conserving cellular automata with one kind of particle.

The output of both methods is a "flow function", which describes the movement of the particles. In the first method, one puts increasingly stronger restrictions on the particle flow until a single flow function is specified. There are no dead ends, every choice of restriction steps ends with a flow.

The second method uses the fact that the flow functions can be ordered and then form a lattice. This method consists of a recipe for the slowest flow that enforces a given minimal particle speed in one given neighbourhood. All other flow functions are then maxima of sets of these flows.

Other questions, like that about the nature of non-deterministic number-conserving rules, are treated briefly at the end.

1. Introduction

Cellular automata are microscopic worlds: extremely simple spaces in which time passes and events occur. Sometimes they are used to simulate aspects of this world. It is therefore an interesting question to ask which of these micro-worlds can be interpreted as containing particles that can move, collide or stick to each other, but are neither destructed nor created.

*Address for correspondence: markus2.redeker@mail.de

Received July 2021; accepted August 2022.

In this paper I provide a new answer to this question, valid for one-dimensional automata with only one kind of particles. The new idea is that such cellular automata are now *constructed* instead of testing whether a given system satisfies the requirements.

The cellular automata are constructed in terms of *flow functions*. These are functions that describe how many particles cross the boundary between two cells, depending on the neighbourhood of this boundary. By specifying the flow functions, one can find all number-conserving automata.

Flow functions did already occur in the literature. They were described by Hattori and Takesue [1, Th. 2.2] and by Pivato [2, Prop. 12] but not used for the construction of transition rules for cellular automata. Imai and Alhazov [3] have defined flow functions for cellular automata of radius $\frac{1}{2}$ and 1, and their Proposition 3 contains a necessary and sufficient condition for number-conservation in radius $\frac{1}{2}$ cellular automata – it is the case of $\ell = 1$ for Theorem 4.3 of this paper, so to speak.

This paper contains two related constructions for flow functions (and therefore number-conserving cellular automata). The first one works by stepwise refinement. It starts with very weak requirements on the flow function, which then are successively strengthened until a single function remains that satisfies all requirements. As a side effect, the construction gives an overview over the set of all number-conserving cellular automata; this could be used for e. g. the classification or enumeration of the automata.

To write about particle flows, one needs an appropriate notation. It should represent all flows and only them, and ideally it should relate in a meaningful way to the represented flows. So far, the flows were represented by functions from cellular neighbourhoods to the integers, which violated the second requirement because flow functions cannot easily be distingushed from non-flow functions.

To correct this, a second construction is introduced, based on the first one. It uses the fact that the flow functions form a lattice and the minimal elements of this lattice have a reasonably simple form: An arbitrary flow function can therefore always written as the maximum of some of these minimal flows.

Background The following paragraphs contain references to papers that are somewhat related to this work. The list is by no means complete. For a little bit of history of number conservation see Bhattacharjee *et al.* [4, Sec. 4.6] or the introduction of the related paper by Wolnik *et al.* [5].

The most important predecessor of the current work is the paper of Hattori and Takesue [1] about additive invariants of cellular automata. The number of particles is one of them. Hattori and Takesue found a simple way to test whether a one-dimensional cellular automaton is number-conserving. This method works only *a posteriori*, but it provides the base for most later papers about number conservation.

Boccara and Fukś [6] describe the behaviour of number-conserving cellular automata with the help of “motion representation” diagrams. These are diagrams that show the motion of individual particles in finite regions of the automaton. The authors find a set of equations for the transition rule of a number-conserving automaton and solve them. They also show that number-conserving rules can be identified by verifying that the number of particles is conserved on all circular cell configurations of a specified size.

Fukś [7] shows that in one-dimensional number-conserving automata, one can assign permanent identities to the “particles” and describe the evolution of the cellular automaton in terms of particle movements.

Pivato [2] derives several characterisations of conservation laws in the more general context of cellular automata on arbitrary groups.

Durand, Formenti and Róka [8] collect different definitions of number conservation and show that they are equivalent. They also derive conditions for number conservation in cellular automata of any dimension.

Overview The rest of this article, after the introduction and a section with definitions, consists of the following major parts.

In Section 3, the *flow function* for a number-conserving cellular automaton is defined and the *flow conditions* are derived: conditions on the flow functions that are satisfied if and only if the functions belong to a number-conserving cellular automaton.

In Section 4, all solutions to the flow conditions are found. Examples and a diagram notation for the flow function follow in the next section.

In Section 6, a lattice structure for the set of flow functions is found. This leads to the notion of a set of "minimal" flow functions from which all other flows can be built with the help of lattice operations. A recipe for minimal flow functions is found and examples for minimal flows are shown.

Two sections, one about related topics and one with open questions, follow at the end.

2. Definitions

A *one-dimensional cellular automaton* is a discrete dynamical system; its states are configurations of simpler objects, the *cells*. The cells are arranged in an infinite line – they are indexed by $\mathbb{Z}$ – and the state of each cell is an element of a finite set Σ.

2.1. Cells and states

In a number-conserving cellular automaton, we picture each cell as a container for a certain number of particles; the number of particles it contains is part of its state. Since the number of states is finite, there is a maximal number C of particles that a cell may contain, the *capacity* of the cellular automaton. We therefore have for each cell state $\alpha \in \Sigma$ a number $\#\alpha \in \{0, \dots, C\}$, the *particle content* of α. The expression

$$\#^c\alpha = C - \#\alpha \tag{1}$$

stands for the *complement* of the particle content: It is the maximal number of particles one can still put into a cell of state α without exceeding its capacity.

To make things simpler, the mapping $\#\colon \Sigma \to \{0, \dots, C\}$ is required to be surjective. On the other hand, it is not necessary that $\#$ is injective; the constructions in this paper will work even if this is not the case.

Sometimes we will take the numbers $\{0, \dots, C\}$ directly as cell states. Such a state set is called a *minimal set* of states. Even then, we will keep the distinction between α and $\#\alpha$, to make it clearer whether we are speaking of cells or of the number of particles in them.

2.2. Configurations

We will need to speak about finite and double infinite sequences of cell states. Finite sequences are also called *strings*, while the infinite sequences are *configurations*.

A *string* is an element of $\Sigma^* = \bigcup_{k \geqslant 0} \Sigma^k$. If $a \in \Sigma^k$ is a string, then k is its *length*, and we write $|a|$ for the length of a. For the string of length 0 we write ϵ.

Often we write a string $a \in \Sigma^k$ as a product of cell states, in the form

$$a = a_0 \dots a_{k-1} \,. \tag{2}$$

The number of particles in a and its complement are then

$$\#a = \sum_{i=0}^{k-1} \#a_i \qquad \text{and} \qquad \#^c a = |a|C - \#a \,. \tag{3}$$

We will also need substrings of a. They are specified by start point and length, in the form

$$a_{m:n} = a_m \dots a_{m+n-1} \,. \tag{4}$$

A *configuration* of a cellular automaton is a doubly infinite sequence of cell states, in the form

$$a = \dots a_{-3} a_{-2} a_{-1} a_0 a_1 a_2 a_3 \dots \,. \tag{5}$$

The conventions for substrings and particle content are the same for configurations as for strings, except that $\#a$ is only defined when the number of the a_i that have a non-zero content is finite. Such a configuration is said to have a *finite particle content*.

2.3. Evolution

The configuration of a cellular automaton changes over time. This behaviour – the *evolution* of the automaton – is specified by the *global transition rule* $\hat{\varphi}$, which maps the configuration at time t to the configuration at time $t + 1$.

As a function, $\hat{\varphi}$ is determined by two integers r_1 and r_2 and a *local transition rule* $\varphi \colon \Sigma^{r_1+r_2+1} \to \Sigma$, subject only to the condition that $r_1 + r_2 \geqslant 0$. The numbers r_1 and r_2 are the *left* and *right radius* of φ. (Almost always we will also require that r_1 and r_2 are non-negative numbers. The only exceptions are the shift rules in the example below.)

Now we can express the value $\hat{\varphi}(a)$ for an arbitrary configuration a by the condition that

$$\hat{\varphi}(a)_x = \varphi(a_{x-r_1}, \dots, a_{x+r_2}) \quad \text{for all } x \in \mathbb{Z}. \tag{6}$$

If we want to use the colon form for substrings that is defined in (3), we can write this rule also in the form $\hat{\varphi}(a)_x = \varphi(a_{x-r_1:r_1+r_2+1})$. This form, which emphasises the length of the local neighbourhood but is less symmetric, will soon be used.

(Often one only considers the symmetric case with $r_1 = r_2 = r$. Then r is called the *radius* of the transition rule.)

2.4. Rules with equivalent dynamics

When we create a new transition rule φ_v by replacing the radii r_1 and r_2 of φ with $r_1' = r_1 + v$ and $r_2' = r_2 - v$, the behaviour of the new transition rule $\hat{\varphi}_v$ does not differ significantly from that of $\hat{\varphi}$, except for a horizontal shift that occurs at each time step.

It has then the dynamics $\hat{\varphi}_v(a)_x = \varphi(a_{x-r_1-v:r_1+r_2+1})$, or equivalently,

$$\hat{\varphi}_v(a)_{x+v} = \varphi(a_{x-r_1:r_1+r_2+1}) \qquad \text{for all } x \in \mathbb{Z}. \tag{7}$$

At every time step, the rule $\hat{\varphi}_v$ moves therefore the new cell states v positions more to the right than $\hat{\varphi}$. We can say that $\hat{\varphi}_v$ is $\hat{\varphi}$ as seen by an observer who moves with speed v to the left.

All rules φ_v can be treated in essentially the same way, independent of the value of v. We therefore introduce a new parameter $\ell = r_1 + r_2$ to characterise φ. The number ℓ is, anticipatingly, called the *flow length* of φ.[1]

Shift rules The simplest example for rules with the same dynamics are the *shift rules* $\hat{\sigma}^k$, which exist for all $k \in \mathbb{Z}$. The rule $\hat{\sigma}^k$ moves in each time step the states of all cells by k positions to the right.

They have $r_1 = k$, $r_2 = -k$ (therefore $\ell = 0$) and a local transition rule of the form $\sigma^k \colon \Sigma \to \Sigma$. Since all shift rules have the same dynamics, all σ^k must be the same function: It is the identity. All $\hat{\sigma}^k$ are trivially number-conserving, independent of the particle contents of the cell states.

This is not the only representation of the shift rules as cellular automata. We will soon see another representation for the rules $\hat{\sigma}^k$ with $k \geqslant 0$.

3. The flow conditions

A formal definition for number conservation is still missing. We use this one:

> A transition rule $\hat{\varphi}$ is *number-conserving* if $\#\hat{\varphi}(a) = \#a$ for every configuration a with finite particle content.

From now on we will therefore assume that every configuration has a finite particle content.

If we divide such a configuration into two parts, we can define the *particle flow* between them. It is the number of particles that move from the left to the right side under the action of $\hat{\varphi}$.

For the definition, consider the following setup:

$$\dots u_{-3}u_{-2}u_{-1} \,\big|\, v_0v_1v_2 \dots . \tag{8}$$

It shows the cells of a configuration, split by the vertical bar into a left part, u, and a right part, v. Its image under $\hat{\varphi}$ shall be

$$\dots u'_{-3}u'_{-2}u'_{-1} \,\big|\, v'_0v'_1v'_2 \dots . \tag{9}$$

[1] At this point, a parameter with a value of $r_1 + r_2 + 1$ might look more natural, but we will see that $r_1 + r_2$ occurs more often in our calculations.

Let now $L = \#u' - \#u$ be the change in the particle content in the left part and $R = \#v' - \#v$ be the change in the right part. The particle content in the configuration is preserved, therefore is $R = -L$. The number R is then the particle flow; if it is positive, particles move to the right.

We apply this concept to the following setup, which is the same as (8), but with the cells named differently.

$$\dots x_{-2}x_{-1}\, u_0 \dots u_{r_1-1} \,\big|\, u_{r_1} \dots u_{\ell-1}\, y_\ell y_{\ell+1} \dots . \tag{10}$$

The two parts are separated by a vertical bar as before, but but there are now also three regions, x, u, and y. Region u consists of r_1 cells to the right and r_2 cells to the left of the vertical bar. The numbers L and R are defined as before.

The interaction in a cellular automaton is local, therefore L can only depend on x and u, and R only on u and y. But $R = -L$, therefore they both can only depend on u. So we can find a function

$$f\colon \Sigma^\ell \to \mathbb{Z} \tag{11}$$

with $f(u) = R = -L$. This is the function which will allow us to describe the essential properties of a number-conserving cellular automaton. We call f the *flow function* of φ.

Next we try to reconstruct φ from f. To do this, we consider the following setup with $\ell + 1$ named cells and two boundaries,

$$\dots w_0 \dots w_{r_1-1} \,\big|\, w_{r_1} \,\big|\, w_{r_1+1} \dots w_\ell \dots . \tag{12}$$

We are interested in the state of the cell at the centre in the next time step. Initially, it contains $\#w_{r_1}$ particles. One step later, there are $\#\varphi(w)$ particles at this place. On the other hand, during this transition, $f(w_{0:\ell})$ particles must have entered the cell region through the left boundary, and $f(w_{1:\ell})$ of them must have left it to the right. The number of particles at the centre at the next time step must therefore be

$$\#\varphi(w) = f(w_{0:\ell}) + \#w_{r_1} - f(w_{1:\ell})\,. \tag{13}$$

With this equation, applied to all neighbourhoods $w \in \Sigma^{\ell+1}$, we can partially reconstruct φ from f. If the state set Σ is minimal, the transition function can even be reconstructed uniquely from the values of $\#\varphi(w)$. Otherwise, there are several different transition functions for the same flow function. (How they are related would be the subject of another paper.)

Since φ can be derived from f, it will be enough to consider f alone. We therefore need to find all functions $f\colon \Sigma^\ell \to \mathbb{Z}$ for which there is a valid transition rule φ. These are exactly those functions f for which the right side of (13) is neither too small nor too large. We have therefore

Theorem 3.1. (Flow conditions)
A number-conserving cellular automaton for a given function $f\colon \Sigma^\ell \to \mathbb{Z}$ exists if and only if

$$0 \leqslant f(w_{0:\ell}) + \#w_{r_1} - f(w_{1:\ell}) \leqslant C \qquad \text{for all } w \in \Sigma^{\ell+1}. \tag{14}$$

The inequalities (14) are called the *flow conditions*.

4. Solving the flow conditions

Great flows have little flows next to them to guide 'em,
And little flows have lesser flows, but not ad infinitum.

We will now restrict out work to rules with $r_1 = \ell$ and $r_2 = 0$, so that particles only move to the right. This will make induction on the neighbourhood size much simpler. It does not reduce the generality of the conclusions because, as we have seen in Section 2.4, every rule is equivalent to such a rule. And at the end, in Section 7.2, we will return to the general case.

The flow conditions now have the form

$$0 \leqslant f(w_{0:\ell}) + \#w_\ell - f(w_{1:\ell}) \leqslant C \qquad \text{for all } w \in \Sigma^{\ell+1}. \tag{15}$$

To find solutions for them, we define some functions related to f, the *half-flows*. They come in two variants, as *bound* and *free* half-flows, where the free half-flows are a generalisation of the bound half-flows.

The bound half-flows have their name from the fact that they are constructed from a specific flow function f. This makes their definition easier to understand and therefore they are introduced first. Properties of the bound half-flows will then lead to a definition for free half-flows, which do not refer to a flow function. Instead, each system of free half-flows is used to *construct* a flow function.

Bound half-flows We will now introduce the bound half-flows. There are two families of them, the *lower* and the *upper half-flows*, given by the equations,

$$\underset{\sim}{f}_k(v) = \min\{\, f(uv) \colon u \in \Sigma^{\ell-k} \,\}, \tag{16a}$$

$$\widetilde{f}_k(v) = \max\{\, f(uv) \colon u \in \Sigma^{\ell-k} \,\}, \tag{16b}$$

with $0 \leqslant k \leqslant \ell$ and $v \in \Sigma^k$. (Note that $\widetilde{f}_0$ and $\underset{\sim}{f}_0$ are in fact constants.) Much more useful are however the inductive forms of these definitions,

$$\underset{\sim}{f}_\ell(w) = f(w), \qquad \underset{\sim}{f}_k(v) = \min\{\, \underset{\sim}{f}_{k+1}(\alpha v) \colon \alpha \in \Sigma \,\}, \tag{17a}$$

$$\widetilde{f}_\ell(w) = f(w), \qquad \widetilde{f}_k(v) = \max\{\, \widetilde{f}_{k+1}(\alpha v) \colon \alpha \in \Sigma \,\}, \tag{17b}$$

with $w \in \Sigma^\ell$, $\alpha \in \Sigma$ and $v \in \Sigma^k$ for $1 \leqslant k \leqslant \ell$.

Example: Flow functions for shift rules The shift rules $\hat{\sigma}^\ell$ for $\ell \geqslant 0$ illustrate these definitions. Each shift rule has a flow function $f \colon \Sigma^\ell \to \{0, \dots, \ell C\}$ with $f(v) = \#v$ for all v. Its half-flows are $\underset{\sim}{f}_k(w) = \#w$ and $\widetilde{f}_k(w) = (\ell - k)C + \#w$, for $w \in \Sigma^k$. This shows that the particle flow can be quite large, greater than the capacity of a single cell.

4.1. Properties of the bound half-flows

The most important consequence of (17) is the following lemma, which shows how the half-flows are related to each other.

It especially shows that for $1 \leqslant k \leqslant \ell$, the flow conditions (15) split into pairs of inequalities,

$$0 \leqslant \underset{\sim}{f}_k(w_{0:k}) + \#w_k - \underset{\sim}{f}_k(w_{1:k}), \tag{18a}$$

$$\widetilde{f}_k(w_{0:k}) + \#w_k - \widetilde{f}_k(w_{1:k}) \leqslant C. \tag{18b}$$

In the lemma, these inequalities follow from in (19): We can get them by removing the central terms of (19a) and (19b) and rearranging the remaining inequalities.

Lemma 4.1. (Interaction of half-flows)
Let $0 \leqslant k < \ell$ and $w \in \Sigma^{k+1}$. If f is the flow function of a number-conserving cellular automaton, then

$$\underset{\sim}{f}_k(w_{1:k}) \leqslant \underset{\sim}{f}_{k+1}(w) \leqslant \underset{\sim}{f}_k(w_{0:k}) + \#w_k, \tag{19a}$$

$$\widetilde{f}_k(w_{0:k}) - \#^c w_k \leqslant \widetilde{f}_{k+1}(w) \leqslant \widetilde{f}_k(w_{1:k})\,. \tag{19b}$$

Proof:
The two pairs of inequalities can be proved independently of each other, so we begin with (19a).

Its proof is a finite induction from $k = \ell$ down to $k = 1$. The induction step consists of showing that that from

$$\underset{\sim}{f}_k(w_{1:k}) \leqslant \underset{\sim}{f}_k(w_{0:k}) + \#w_k \qquad \text{for all } w \in \Sigma^{k+1} \tag{20}$$

always follows

$$\underset{\sim}{f}_{k-1}(w'_{1:k-1}) \leqslant \underset{\sim}{f}_k(w') \leqslant \underset{\sim}{f}_{k-1}(w'_{0:k-1}) + \#w'_{k-1} \qquad \text{for all } w' \in \Sigma^k. \tag{21}$$

The induction can begin because for $k = \ell$, inequality (20) is equivalent to the left side of the flow condition (15).

To prove the induction step, note that the left side of (20) is independent of the choice of w_0. At the right side of (20), we can therefore replace the term $\underset{\sim}{f}_k(w_{0:k})$ with the smallest possible value it can take when we vary w_0 and keep $w_{1:k-1}$ fixed. The result is $\underset{\sim}{f}_{k-1}(w_{1:k-1})$, and we get $\underset{\sim}{f}_k(w_{1:k}) \leqslant \underset{\sim}{f}_{k-1}(w_{1:k-1}) + \#w_k$. This is already the right inequality of (21); we only need to write $w_{1:k}$ as w'. The left inequality, $\underset{\sim}{f}_{k-1}(w'_{1:k-1}) \leqslant \underset{\sim}{f}_k(w')$, follows directly from the inductive definition (17a) of $\underset{\sim}{f}_k$.

The proof of (19b) is similar. We only need to replace $\underset{\sim}{f}$ with $\widetilde{f}$, $\#\beta$ with $-\#^c\beta$ and revert the order of the terms in the inequalities. □

The next lemma is about the behaviour of $\underset{\sim}{f}_k(v)$ and $\widetilde{f}_k(v)$ for a single value of v.

Lemma 4.2. (Bounds on the half-flows)
Let f be the flow function of a number-conserving cellular automaton. Then

$$0 \leqslant \underset{\sim}{f}_k(v) \leqslant \#v, \tag{22a}$$

$$0 \leqslant \widetilde{f}_k(v) \leqslant (\ell - k)C + \#v \tag{22b}$$

and

$$\underset{\sim}{f}_k(v) \leqslant \widetilde{f}_k(v) \leqslant \underset{\sim}{f}_k(v) + (\ell - k)C \tag{23}$$

are valid for $0 \leqslant k \leqslant \ell$ and $v \in \Sigma^k$.

Proof:
The first two inequalities follow directly from the definitions of $\underset{\sim}{f}_k(v)$ and $\widetilde{f}_k(v)$ in (16). For both half-flows, we have to consider all flows $f(uv)$ with $u \in \Sigma^{\ell-k}$, i. e. the following setup:

$$\ldots u_1 \ldots u_{\ell-k} v_1 \ldots v_k \,\big|\, \ldots \tag{24}$$

Only the particles in u and v may cross the boundary, and they may only move to the right. All possible values for $f(uv)$ are therefore non-negative, and the same is true for $\underset{\sim}{f}_k(v)$ and $\widetilde{f}_k(v)$. This proves the two lower bounds in (22). For the upper bound on $\underset{\sim}{f}_k(v)$, we note that the set of all $f(uv)$ includes the case with $\#u = 0$. Then at most $\#v$ particles may cross to the right, which means that also $\underset{\sim}{f}_k(v) \leqslant \#v$. On the other hand it is also possible that all cells in u contain C particles and that $\#u = (l - k)C$. Then $(\ell - k)C + \#v$ particles may cross the boundary, which explains the upper bound for $\widetilde{f}_k(v)$.

The left side of (23) is clear; the right side can be proved with another setup:

$$\ldots v_1 \ldots v_k \,\big|\, w_1 \ldots w_{\ell-k} \,\big|\, \ldots \tag{25}$$

Here we assume that we know $f(vw)$ and try to find upper and lower bounds for the number of particles that cross the left boundary. The highest possible number of particles is $f(vw) + \#^c w$, because then the w region will be completely filled up in the next time step. The smallest particle flow is $f(vw) - \#w$, because then the w region will be empty. This means that $\widetilde{f}_k(v) \leqslant f(vw) + \#^c w$ and $\underset{\sim}{f}_k(v) \geqslant f(vw) - \#w$. When we subtract these inequalities, we get $\widetilde{f}_k(v) - \underset{\sim}{f}_k(v) \leqslant \#^c w + \#w = (\ell - k)C$, from which the right side of (23) follows. □

4.2. The construction of all flows

Now we can define the free half-flows. They provide us with a practical method to find the flow functions among all the functions $f\colon \Sigma^\ell \to \{0, \ldots, \ell C\}$. This method is an algorithm of stepwise refinement, where at each step a choice takes place that never needs to be taken back.

A system of *free half-flows* is then a sequence $(\underline{f}_k, \overline{f}_k)_{0 \leqslant k \leqslant \ell}$ of functions that satisfy the conditions of Lemma 4.1 and 4.2. This means that

$$\underline{f}_k(w_{1:k}) \leqslant \underline{f}_{k+1}(w) \leqslant \underline{f}_k(w_{0:k}) + \#w_k, \tag{26a}$$

$$\overline{f}_k(w_{0:k}) - \#^c w_k \leqslant \overline{f}_{k+1}(w) \leqslant \overline{f}_k(w_{1:k}) \tag{26b}$$

must be true for $0 \leqslant k < \ell$ and $w \in \Sigma^{k+1}$ and

$$0 \leqslant \underline{f}_k(v) \leqslant \#v, \tag{26c}$$

$$0 \leqslant \overline{f}_k(v) \leqslant \#v + (\ell - k)C, \tag{26d}$$

$$\underline{f}_k(v) \leqslant \overline{f}_k(v) \leqslant \underline{f}_k(v) + (\ell - k)C \tag{26e}$$

must be true for $0 \leqslant k \leqslant \ell$ and $v \in \Sigma^k$. (To distinguish between the two kinds of half-flows, we write free half-flows with a bar and bound half-flows a tilde.)

This definition ensures that each system of bound half-flows is also a system of free half-flows and also that free half-flows can be chosen without reference to a given flow function.

The following theorem then states how flow functions are constructed.

Theorem 4.3. (Flow construction)
All solutions of the flow conditions (15) can be found by constructing a sequence of free half-flows

$$\underline{f}_0, \overline{f}_0, \underline{f}_1, \overline{f}_1, \dots, \underline{f}_\ell, \overline{f}_\ell\,. \tag{27}$$

This means that if $j < \ell$ and $\underline{f}_0$, $\overline{f}_0$, $\underline{f}_1$, $\overline{f}_1$, $\dots$, $\underline{f}_j$, $\overline{f}_j$ is a sequence of functions[2] satisfying the conditions in (26), it is always possible to extend it with two functions $\underline{f}_{j+1}$ and $\overline{f}_{j+1}$ so that the new sequence again satisfies (26).

At the end, we have $\underline{f}_\ell = \overline{f}_\ell$ and the function $f = \underline{f}_\ell = \overline{f}_\ell$ is the constructed flow. All flow functions can be found in this way.

Proof:
We have already seen that for every flow there is a sequence of free half-flows that satisfies the required inequalities. Therefore all possible flows can be reached by the construction of the theorem.

It remains to show that the construction can always be completed. We will now follow it step by step and show that at each step is always possible.

The construction starts with the choice of two constants $\underline{f}_0$ and $\overline{f}_0$ that satisfy the conditions in (26). For $k = 0$, the conditions reduce to

$$\underline{f}_0 = 0 \qquad \text{and} \qquad 0 \leqslant \overline{f}_0 \leqslant \ell C, \tag{28}$$

so the first step can always be done.

Now assume that $k < \ell$, that the half-flows $\underline{f}_k$ and $\overline{f}_k$ are already constructed and we want to construct $\underline{f}_{k+1}$ and $\overline{f}_{k+1}$. We must then find solutions for the inequalities (26). I will now rewrite them in the order in which we will try to find solutions for them.

[2]The empty sequence is here included.

The result is the following system of inequalities. For any $v \in \Sigma^{k+1}$ it contains all conditions on $\underline{f}_{k+1}(v)$ and $\overline{f}_{k+1}(v)$.

$$\underline{f}_k(v_{1:k}) \leqslant \underline{f}_{k+1}(v) \leqslant \underline{f}_k(v_{0:k}) + \#v_k, \tag{29a}$$

$$0 \leqslant \underline{f}_{k+1}(v) \leqslant \#v, \tag{29b}$$

$$\overline{f}_k(v_{0:k}) - \#^c v_k \leqslant \overline{f}_{k+1}(v) \leqslant \overline{f}_k(v_{1:k}) \tag{29c}$$

$$0 \leqslant \overline{f}_{k+1}(v) \leqslant \#v + (\ell - k - 1)C \tag{29d}$$

$$\underline{f}_{k+1}(v) \leqslant \overline{f}_{k+1}(v) \leqslant \underline{f}_{k+1}(v) + (\ell - k - 1)C, \tag{29e}$$

There are $|\Sigma|^{k+1}$ of such systems of inequalities, but each pair of $\underline{f}_{k+1}(v)$ and $\overline{f}_{k+1}(v)$ occurs in only one of them. The values of $\underline{f}_{k+1}(v)$ and $\overline{f}_{k+1}(v)$ can therefore be chosen independently for each v.

First we note that each of the requirements in (29) can be satisfied individually. The only cases in which this is not obvious are (29a) and (29c). But when we remove the central term in both, a half-flow condition (18) for $\underline{f}_k$ or $\overline{f}_k$ remains, which is true by induction. The upper and lower bounds on $\underline{f}_{k+1}$ and $\overline{f}_{k+1}$ in each of the two requirements are therefore consistent with each other and we can find solutions for them.

Next we verify that all requirements in (29) except the right inequality in (29e) can be satisfied together. We do that by choosing

$$\underline{f}_{k+1}(v) = \underline{f}_k(v_{1:k}), \qquad \overline{f}_{k+1}(v) = \min\{\overline{f}_k(v_{1:k}), \#v + (\ell - k - 1)C\} \tag{30}$$

as our candidate for a solution. This is the "lazy" solution in which $\underline{f}_{k+1}(v)$ is as small and $\overline{f}_{k+1}(v)$ as large as possible. The first two requirements in (29) are then clearly satisfied. For the next two, we need only to verify the left inequality of (29c). It is equivalent to the two inequalities

$$\overline{f}_k(v_{0:k}) - \#^c v_k \leqslant \overline{f}_k(v_{1:k}), \qquad \overline{f}_k(v_{0:k}) - \#^c v_k \leqslant \#v + (\ell - k - 1)C\,. \tag{31}$$

The left inequality is part of (29c) and true by induction. For the right inequality, we add on both sides $\#^c v_k$ and get $\overline{f}_k(v_{0:k}) \leqslant \#v_{0:k} + (\ell - k)C$. This is the version of (29d) with $v_{0:k}$ instead of v and again true by induction.

What remains is the right inequality in (29e). It may be violated by the solution candidate (30). If that is the case, we move the candidates for $\underline{f}_{k+1}(v)$ and $\overline{f}_{k+1}(v)$ stepwise towards each other until we have either found a full solution or cannot continue. If we cannot continue, $\underline{f}_{k+1}(v)$ cannot be made larger or $\overline{f}_{k+1}(v)$ smaller. That is, one of the left inequalities of (29a) and (29b) and one of the right inequalities of (29c) and (29d) must have become equalities, or $\underline{f}_{k+1}(v) = \overline{f}_{k+1}(v)$.

Three cases can be excluded. If $\underline{f}_{k+1}(v) = \overline{f}_{k+1}(v)$, then all of (29e) is satisfied and we have a solution to the full system of inequalities. If $\overline{f}_{k+1}(v) = 0$, then $\underline{f}_{k+1}(v) = 0$ too and we have again a solution. And $\underline{f}_{k+1}(v) = \#v$ cannot occur without $\underline{f}_{k+1}(v) = \underline{f}_k(v_{0:k}) + \#v_k$ because by induction, $\underline{f}_k(v_{0:k}) \leqslant \#v_{0:k}$ and therefore $\underline{f}_k(v_{0:k}) + \#v_k \leqslant \#v$.

So we must have $\underline{f}_{k+1}(v) = \underline{f}_k(v_{0:k}) + \#v_k$ and $\overline{f}_{k+1}(v) = \overline{f}_k(v_{0:k}) - \#^c v_k$. But then

$$\begin{aligned}\overline{f}_{k+1}(v) &= \overline{f}_k(v_{0:k}) - \#^c v_k \\ &\leqslant \underline{f}_k(v_{0:k}) + (\ell - k)C - C + \#v_k \\ &= \underline{f}_{k+1}(v) + (\ell - k - 1)C\,.\end{aligned}$$

So the right inequality of (29e) can always be satisfied.

In other words, the step from $\underline{f}_k$ and $\overline{f}_k$ to $\underline{f}_{k+1}$ and $\overline{f}_{k+1}$ is always possible, and $\underline{f}_\ell$ and $\overline{f}_\ell$ can always be constructed. That they are equal follows from (23). □

5. Examples and diagrams

We will now, as an illustration for Theorem 4.1, construct all number-conserving elementary cellular automata.

This would lead to a quite voluminous computation if it were done directly. So I will at first introduce a diagram notation for all the flows and half-flows of a number-conserving cellular automaton. With these *box diagrams*, the computation can be shown in a reasonable amount of space.

5.1. A rule with a maximal half-flow

Before we can begin to work with elementary cellular automata, I will illustrate the method by applying it to a simpler example. We will now construct a flow function f for which the half-flow $\overline{f}_0$ takes the largest possible value. We will do this for the minimal state set with capacity $C = 1$, i. e. the set $\Sigma = \{0, 1\}$, so that every cell contains at most one particle. The diagrams for the computation are shown in Figure 1.

The construction starts with the box at the top; that at the bottom represents the resulting flow function. I will first describe how the diagrams must be read and the how the construction proceeds.

- Each box stands for a pair of half-flow values, $\underline{f}_k(w)$ and $\overline{f}_k(w)$ for some w. The box for $\underline{f}_k(w)$ and $\overline{f}_k(w)$ is called the *k-box* for w.

  In the construction, it influences the flow values for all neighbourhoods with w at its right end. In the figure, one can find these neighbourhoods at the bottom, deep below the k-box.

- The values of $\underline{f}_k(w)$ and $\overline{f}_k(w)$ are given by the position of the lower and the upper edge of the k-box for w.

- The values of w for the innermost k-boxes in a diagram appear at its bottom, below the box.

- The dots in the bottom diagram stand for the value of f and at the same time for the values of the half-flows $\underline{f}_\ell$ and $\overline{f}_\ell$, since they all are the same.

  Under each dot stands the neighbourhood which it represents. (The neighbourhoods are ordered lexicographically by their mirror-images.)

- The grey lines in the diagrams are restrictions for the upper and lower edges of the $(k+1)$-boxes that will be constructed in the next step.

The construction itself obeys the following rules, which are a "graphical version" of Theorem 4.3.

- All k-boxes except the outermost one must be drawn directly inside a $(k+1)$-box. This ensures that the conditions $\underline{f}_k(v_{1:k}) \leqslant \underline{f}_{k+1}(v)$ and $\overline{f}_{k+1}(v) \leqslant \overline{f}_k(v_{1:k})$ of (29a) and (29c) are satisfied.
- The maximal height of a k-box is $\ell - k$. This ensures that (29e) is satisfied.
- The upper edge of a k-box is not higher than the greatest particle content of all the neighbourhoods that it influences.

  This ensures condition (29d), because the upper edge of the k-box v stands for $\overline{f}_k(v)$, which is the maximum of all neighbourhoods of the form uv with $u \in \Sigma^{\ell-k}$: Their highest particle content is therefore $(\ell - k)C + \#v$, as required in the condition.
- When a new k-box is constructed, its dimensions are partially determined by the pair of grey lines in the diagram above it that have the same horizontal extension as itself. The new k-box must have an upper edge that is not lower than the upper grey line, and a lower edge that is not higher than the lower grey line.

  This is to make sure that the conditions $\underline{f}_{k+1}(v) \leqslant \underline{f}_k(v_{0:k}) + \#v_k$ and $\overline{f}_k(v_{0:k}) - \#^c v_k \leqslant \overline{f}_{k+1}(v)$ of (29a) and (29c) are satisfied.
- Finally, to keep the promise we just made, we need to draw new grey lines at the right positions in the new diagram. The grey lines that we need to draw are copies of the arrangement of k-boxes that we just have constructed – but with modifications. (a) The first one is that we create C copies of the arrangement and squeeze them horizontally by a factor C^{-1}, so that they again fit into the same space. One will then be above the neighbourhoods that have 0 as its rightmost state, one above those that end with 1, and so on. (b) We now move the arrangements upward by rising amounts: The first copy not at all, the next one by 1, and so on. (c) Then we replace the boxes in all arrangements by pairs of grey lines: the lower edge by a grey line at the same position, but the upper edge by an upper grey line *that is C positions below*.

  To justify these rules, write the inequalities from (29a) and (29c) that were mentioned above in the form

$$\overline{f}_{k+1}(v) \geqslant \overline{f}_k(v_{0:k}) + \#v_k - C, \tag{32a}$$

$$\underline{f}_{k+1}(v) \leqslant \underline{f}_k(v_{0:k}) + \#v_k\,. \tag{32b}$$

  The upper inequality describes the position of the upper grey line, $\overline{f}_{k+1}(v)$, in terms if the upper edge of the k-box for $v_{0:k}$ and other parameters; the lower inequality is about the lower grey line and the lower edge of the new k-box. We can then see that (a) the terms $\overline{f}_k(v_{0:k})$ and $\underline{f}_k(v_{0:k})$, which stand for the boundaries of the existing k-boxes, occur C times, once for each value of v_k, (b) the addition of $\#v_k$ means that each line pair is is shifted upward by this amount, and (c) the subtraction of C moves each upper grey line down by this amount.

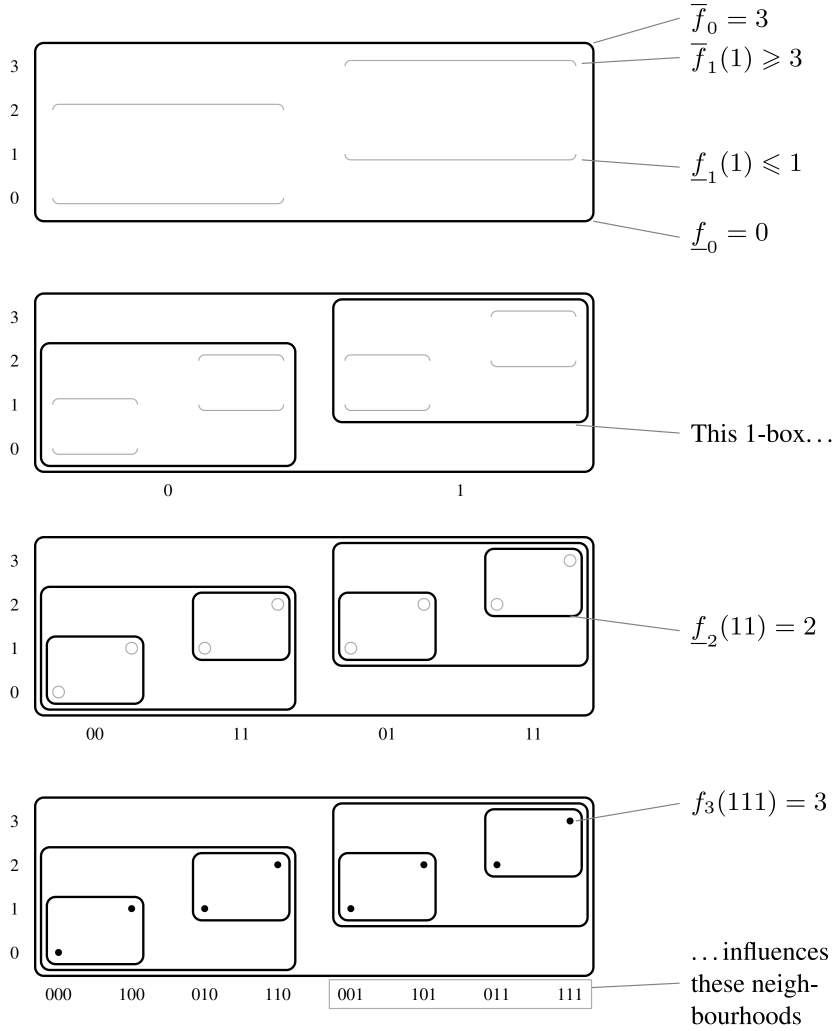


Figure 1. Construction of a flow with $\overline{f}_0 = 3$, with some annotations.

This is the way in which the diagrams in Figure 1 were constructed. As an example for the most difficult part, namely the construction of the thin lines, we will now look at the first diagram in detail. In it, we see two pairs of grey lines that form the "shadows" of two 1-boxes. Both are smaller versions of the outer 0-box with the upper edge lowered by $C = 1$; the right one is also moved upward by 1.

We can also see that in the first three diagrams of Figure 1, the arrangement of grey lines in the right half is a shifted version of the arrangement of grey lines in the right half.

In the diagrams of Figure 1, the pairs of thin lines always have the maximal possible distance, so there is no choice in the following step. This means that the flow function for $\hat{\sigma}^3$ is the only one with $\overline{f} = 3$ – an observation that can be easily extended to all $\hat{\sigma}^\ell$ with $\ell \geqslant 0$.

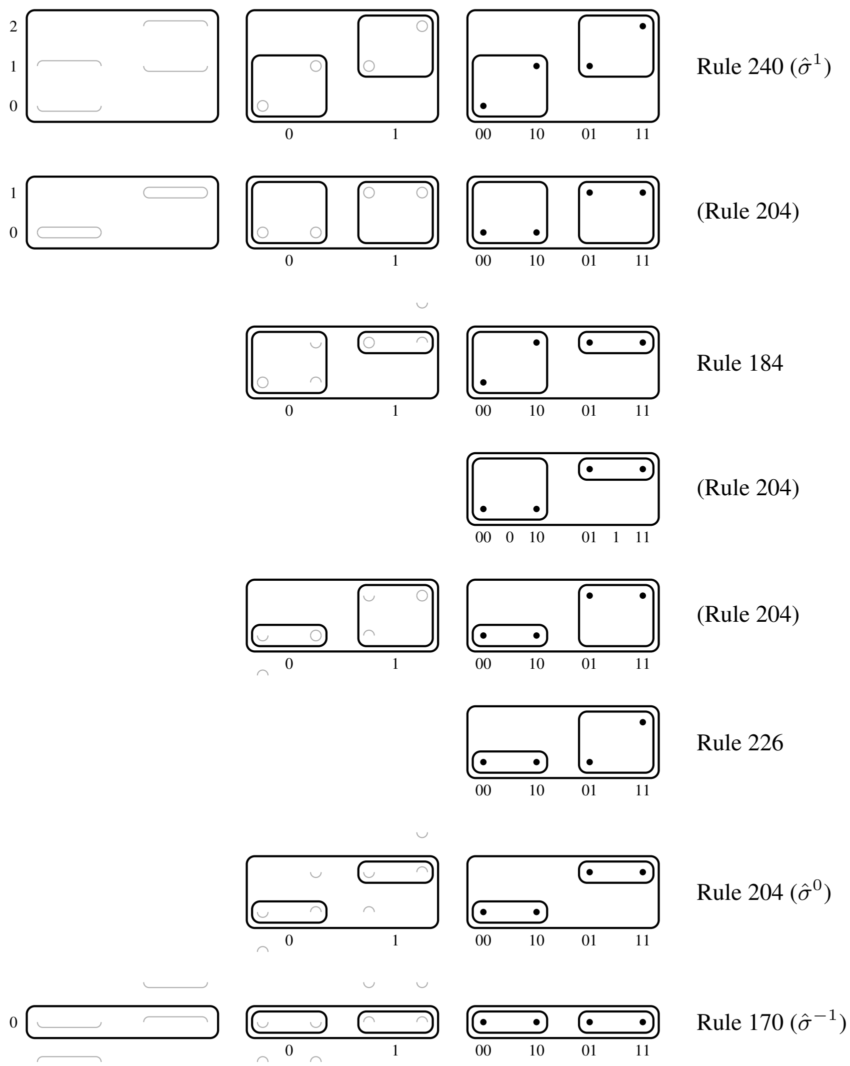


Figure 2. Construction of all number-conserving ECA rules.

5.2. Elementary cellular automata

Now we can begin with the construction of all number-conserving elementary cellular automata. Elementary cellular automata, or ECA [9], are simply the one-dimensional cellular automata with radius $r_1 = r_2 = 1$ and state set $\Sigma = \{0, 1\}$. They therefore have a transition rule $\varphi\colon \Sigma^3 \to \Sigma$ and a flow function of width $\ell = 2$ for capacity $C = 1$. We must now find all number-conserving flows.

This is done in Figure 2. The figure must be read from left to right and from top to bottom. The first column contains all possible box diagrams for the first step of the construction. To the right of each diagram and right-below of it are its refinements, the diagrams for all the possible next steps of the construction. So the second box diagram in the first column has four refinements, and the second of them, two.

The rightmost column contains the code numbers of the ECA that belong to the flows in column 3. The code numbers were given by Wolfram [9]. If a rule is a shift rule $\hat{\sigma}^k$, the name of the rule is also given. Since we have here switched back to a symmetrical cellular neighbourhood, the shift speed is different from that what we would have seen in the context of Figure 1.

Some rule names are put in braces. They belong to rules that occur more than once in the diagram, constructed in different ways. For an unknown reason it is only Rule 204, the identity function, that occurs more than once. But there is only one construction of Rule 204 that uses bound half-flows; it is that one in which the rule name is not put in braces.

One new phenomenon appears here: It can happen that the "upper" grey line is below the "lower" grey line. This occurs the first time in the construction of Rule 184. It is the reason why the two types of grey lines have distinct shapes.

The result of this calculation is that the only number-conserving ECA rules are the shift rules $\hat{\sigma}^{-1}$, $\hat{\sigma}^0$, $\hat{\sigma}^1$, the so-called "traffic rule" 184, and Rule 226, which is Rule 184 with left and right exchanged.

6. Minimal flows

We have now found all flow functions with rule width $\ell = 2$ and for a minimal state set Σ of capacity $C = 1$. Similar computations for larger ℓ or Σ would soon become unwieldy. I will therefore now describe another method to construct all flows for a given ℓ and Σ. It is based on a lattice structure on the set of flows.

The lattice structure also provides formulas that can serve as names for all the flow functions.

6.1. Sets of flows

To keep the following arguments short (or at least not too long), we will introduce some notation. We will write $\mathcal{F}(\ell, \Sigma)$ for the *set of flows* of width ℓ and state set Σ, i. e. the set of functions $f\colon \Sigma^\ell \to \{0, \dots, \ell C\}$, with $C = \max\{\, \#\alpha\colon \alpha \in \Sigma \,\}$ that satisfy the flow conditions (15). (The restriction to $r_1 = \ell$ and $r_2 = 0$ is again in place.)

We will also write $\mathcal{H}(\ell, \Sigma)$ for the *set of free half-flows* related to ℓ and Σ. It is the set of function families $(\underline{f}_k, \overline{f}_k)_{0 \leqslant k \leqslant \ell}$ that satisfy the inequalities (26).

If Σ is a minimal state set of capacity C, we will also write these sets as $\mathcal{F}(\ell, C)$ and $\mathcal{H}(\ell, C)$. When the intended meaning is clear, we may also write $\mathcal{F}$ or $\mathcal{H}$.

6.2. Flows as a partially ordered set

Our next step is the introduction of a partial order on $\mathcal{F}$ and $\mathcal{H}$. We define the order pointwise: For two flows $f, g \in \mathcal{F}$, the relation $f \leqslant g$ shall be true if

$$\forall v \in \Sigma^\ell\colon f(v) \leqslant g(v)\,. \tag{33}$$

For systems of half-flows f, $g \in \mathcal{H}$, conditions similar to (33) must be satisfied for all of their half-flows to make $f \leqslant g$ true.

The *minimum* $f \wedge g$ and *maximum* $f \vee g$ of two flow functions $f, g \in \mathcal{F}$ is defined pointwise too, in the form

$$\forall v \in \Sigma^\ell \colon (f \wedge g)(v) = \min\{f(v), g(v)\}, \tag{34a}$$

$$\forall v \in \Sigma^\ell \colon (f \vee g)(v) = \max\{f(v), g(v)\}\,. \tag{34b}$$

For systems of half-flows, $f \wedge g$ and $f \wedge h$ are defined in the same way – here the formulas of (34) are applied to all pairs of half-flows in f and g with the same type.

The minimum and maximum of a set S of flows (or half-flow systems) is written $\bigwedge S$ and $\bigvee S$. For $S \subseteq \mathcal{F}$ we have then

$$\forall v \in \Sigma^\ell \colon (\bigwedge S)(v) = \min\{\, f(v) \colon f \in S \,\}, \tag{35a}$$

$$\forall v \in \Sigma^\ell \colon (\bigvee S)(v) = \max\{\, f(v) \colon f \in S \,\}, \tag{35b}$$

and again the formulas for half-flows are similar.

The following theorem shows that these definitions are useful.

Theorem 6.1. With the operations $\wedge$ and $\vee$ as defined above, the sets $\mathcal{F}(\ell, \Sigma)$ and $\mathcal{H}(\ell, \Sigma)$ each form a distributive lattice.

Proof:
We will prove the theorem for $\mathcal{F}$ and then sketch a similar argument for $\mathcal{H}$.

First we note that the set $\{0, \dots, C\}^{\Sigma^\ell}$ of functions from Σ^ℓ to $\{0, \dots, C\}$ is a distributive lattice. This is because $\{0, \dots, C\}$, as a linear order, is distributive, and $\{0, \dots, C\}^{\Sigma^\ell}$, as a product of distributive lattices, is so too [10, Proposition 4.8].

It is therefore enough to show that $\mathcal{F}$ is closed under minimum and maximum; then it is a sublattice of the full function space and also distributive [10, Section 4.7].

To do this, let f, $g \in \mathcal{F}$ be two flows and $h = f \wedge g$. We have then to verify that the following inequalities are true for all $v \in \Sigma^\ell$ and $w \in \Sigma^{\ell+1}$:

$$h(v) \geqslant 0, \tag{36a}$$

$$h(v) \leqslant \#v, \tag{36b}$$

$$h(w_{1:\ell}) - h(w_{0:\ell}) \leqslant \#w_\ell, \tag{36c}$$

$$h(w_{0:\ell}) - h(w_{1:\ell}) \leqslant \#^c w_\ell\,. \tag{36d}$$

The first two inequalities control the upper and lower bounds on h and are clearly satisfied. The other two represent the flow conditions (15) for h. We first look at (36c). The two flow values at its left side each stand either for a value of f or of g. If both belong to the same flow, say f, then (36c) is actually the same inequality for that flow and therefore true. The only interesting case is therefore that in which the two h values stem from different flows. Without loss of generality we may assume that the first one is from f and the second from g, so that we need to find an upper bound for $f(w_{1:\ell}) - g(w_{0:\ell})$. But since $h(w_{1:\ell}) = f(w_{1:\ell})$, we must have $f(w_{1:\ell}) \leqslant g(w_{1:\ell})$ and can calculate

$$f(w_{1:\ell}) - g(w_{0:\ell}) \leqslant g(w_{1:\ell}) - g(w_{0:\ell}) \leqslant \#w_\ell, \tag{37}$$

so (36c) must be true. Inequality (36d) has the same form and can be proved in the same way. This proves that $f \wedge g \in \mathcal{F}$.

When we set instead $h = f \vee g$, the proof of (36c) is a bit different. Now we conclude instead from $h(w_{0:\ell}) = g(w_{0:\ell})$ that $g(w_{0:\ell}) \geqslant f(w_{0:\ell})$, and (37) becomes

$$f(w_{1:\ell}) - g(w_{0:\ell}) \leqslant f(w_{1:\ell}) - f(w_{0:\ell}) \leqslant \#w_\ell\,. \tag{38}$$

The rest of the argument is the same, and we have now proved that $\mathcal{F}$ is a distributive lattice.

The space $\mathcal{H}$ of free half-flows is a subset not of a single function space but the product of several, one for each half-flow. This product is still a distributive lattice. Therefore it again is enough to check whether it is closed under $\wedge$ and $\vee$. There are a lot more inequalities to consider, but they all can be brought to one of the three forms

$$h_k(w) \leqslant K, \tag{39a}$$

$$h_k(w) \geqslant K, \tag{39b}$$

$$h_j(v) - h_k(w) \leqslant K\,. \tag{39c}$$

In these inequalities, the h terms stand either for $\underline{h}$ or $\overline{h}$, with possibly different choices in the same inequality,[3] K is a constant that does not depend on the half-flow functions, and $v \in \Sigma^j$ and $w \in \Sigma^k$. These inequalities have the same form as those for $\mathcal{F}$, therefore the same arguments can be used, and $\mathcal{H}$ too is a distributive lattice. □

6.3. Minimal flows as building blocks for all the flows

Now, with the lattice structures of $\mathcal{F}$ and $\mathcal{H}$, we can use a subset of all flows as building blocks for the rest. These are the *minimal flows*

$$m(v,k) = \bigwedge\{\, f \in \mathcal{F} \colon f(v) \geqslant k \,\} \tag{40}$$

with $v \in \Sigma^\ell$ and $k \in \{0, \dots, \#v\}$. So $m(v,k)$ is the smallest flow that at the neighbourhood v has at least the strength k.

Every flow can then be represented as a maximum of minimal flows,

$$f = \bigvee\{\, m(v, f(v)) \colon v \in \Sigma^\ell \,\}\,. \tag{41}$$

[3] Different choices are needed because the sequence $(\overline{h}_k, \underline{h}_k)_k$ must also satisfy the right inequality of (26e), which here becomes $\overline{h}_k(v) \leqslant \underline{h}_k(v) + (\ell - k)C$. To bring it to the form (39c), it must be written as $\overline{h}_k(v) - \underline{h}_k(v) \leqslant (\ell - k)C$.

To see why this is so, we note that $m(v, f(v))$ is a flow that is less than or equal than f and agrees with f when evaluated at v. This is true because for $m(v, f(v))$, the right side of (40) becomes $\bigwedge\{\, g \in \mathcal{F} \colon g(v) \geqslant f(v) \,\}$. This expression is the minimum of a set of flows which contains f, therefore $m(v, f(v)) \leqslant f$. And it is the minimum of a set of functions whose values at v are greater or equal to $f(v)$ and which contains f, therefore $m(v, f(v))(v) = f(v)$.

The first fact proves that the right side of (41) is less than or equal to f; the second fact, that for each neighbourhood v, the right side of (41) has a value that is not less than $f(v)$. So the right side of (41) must be f.

Apart from being building blocks, the minimal flows are also interesting in their own right. They answer the questions: If I require that $f(v) = k$, which influence has this on other neighbourhoods? Does pushing the particles forward in one place set particles in other place in motion? We will therefore construct the minimal flows.

Theorem 6.2. Let $f = m(a, k)$ be a minimal flow in $\mathcal{F}(\ell, \Sigma)$ and $b \in \Sigma^\ell$. Then $f(b)$ is the smallest non-negative number which satisfies the inequality

$$f(b) \geqslant k - \min\{|u|, |u'|\}C - \#w - \#^c w', \tag{42}$$

for all u, v, w, u', $w' \in \Sigma^*$ with $a = uvw$ and $u'vw' = b$.

Proof:
According to Theorem 4.3, $m(a, k)$ can be constructed from free half-flows. We take now the minimum of all half-flow sequences that lead to $m(a, k)$, i. e. the system

$$M(a, k) = \bigwedge\{\, (\underline{f}_k, \overline{f}_k)_k \in \mathcal{H} \colon \underline{f}_\ell(a) = \overline{f}_\ell(a) \geqslant k \,\}. \tag{43}$$

It consists of a sequence of half-flows that leads to $m(a, k)$ "greedily": $M(a, k)$ is a sequence $(\underline{f}_i, \overline{f}_i)_{0 \leqslant i \leqslant \ell}$ of half-flow pairs in which every value $\underline{f}(v)$ and $\overline{f}(v)$ is the smallest possible for which the construction sequence can still end in $m(a, k)$. We will now construct such a sequence.

To do this, we will use the inequalities of (26), but in a much more compressed form. In a first simplification, we write them as

$$\underline{f}(w) \leqslant \underline{f}(vw) \leqslant \underline{f}(v) + \#^c w, \tag{44a}$$

$$\overline{f}(v) - \#^c w \leqslant \overline{f}(vw) \leqslant \overline{f}(w), \tag{44b}$$

$$\underline{f}(v) \leqslant \overline{f}(v) \leqslant \underline{f}(v) + (\ell - |v|)C, \tag{44c}$$

and they are valid for all $v, w \in \Sigma^*$ with $|vw| \leqslant \ell$.

These inequalities differ from their representation in (26) in two aspects: (a) The indices on $\underline{f}$ and $\overline{f}$ are dropped, since they can be derived from their arguments, and writing them would make the formulas only more complicated. (b) In the first two lines, the formulas have been *iterated* and the variables *renamed*. From the inequality $\underline{f}_{k+1}(w) \leqslant \underline{f}_k(w_{0:k}) + \#w_k$ in (19a), valid for $w \in \Sigma^{k+1}$, we get by induction $\underline{f}_{k+n}(w) \leqslant \underline{f}_k(w_{0:k}) + \#w_{k:n}$ for $w \in \Sigma^{k+n}$, and then, after renaming $w_{0:k}$ and $w_{k:n}$ to v and w, the right side of (44a). The other derivations are similar.

You may also have noticed that conditions (26c) and (26d) have disappeared. They are less important. Their left parts state that all half-flows must be non-negative, which we will keep in mind, while their right parts contain upper bounds to the half-flows. We will not need them because in our construction, all half-flows are as small as possible.

In a second compression step, we now express the inequalities (44) with arrows. We write an inequality

$$\underline{f}(u) + n \leqslant \overline{f}(v), \tag{45}$$

as an arrow

$$\underline{u} \xrightarrow{n} \overline{v}, \tag{46}$$

and do the same for all other combinations of upper and lower bars. The number n on an arrow is called its *strength*, and it is omitted when $n = 0$.

An arrow as in (46) can be interpreted as "if $\underline{f}(u) \geqslant x$, then $\overline{f}(v) \geqslant x + n$". So when we have a chain of arrows, like $\overline{u} \xrightarrow{m} \overline{v} \xrightarrow{n} \underline{w}$ we can add their strengths and get a new arrow, in this case $\overline{u} \xrightarrow{m+n} \underline{w}$. We want to find arrows of the form $\underline{a} \xrightarrow{n} \underline{b}$, because they lead to lower bounds on $f(b)$. (Note that, since $|a| = |b| = \ell$, we have $\underline{f}(a) = \overline{f}(a)$ and $\underline{f}(b) = \overline{f}(b)$, so that upper and lower bars on a and b are interchangeable.)

When we now translate the inequalities of (44) into arrows, they become

$$\underline{w} \to \underline{vw}, \qquad \underline{vw} \xrightarrow{-\#w} \underline{v}, \tag{47a}$$

$$\overline{v} \xrightarrow{-\#^c w} \overline{vw}, \qquad \overline{vw} \to \overline{w}, \tag{47b}$$

$$\underline{v} \to \overline{v}, \qquad \overline{v} \xrightarrow{(|v|-\ell)C} \underline{v}\,. \tag{47c}$$

In principle, we must consider all arrow chains from a to b. But most of them can be replaced by stronger chains, and only two remain. The following arguments show how this is done.

1. *We can ignore all chains in which two arrows of the same form occur in sequence.*

   An example for such a chain is $\underline{vww'} \xrightarrow{-\#w'} \underline{vw} \xrightarrow{-\#w} \underline{v}$, in which the left arrow of (47a) occurs twice. It can be replaced with the equally strong arrow $\underline{vww'} \xrightarrow{-\#ww'} \underline{v}$. The same can be done with the other arrows in (47a) and (47b), which are the only arrows to which this rule applies.

2. In the next reduction step, we consider arrow chains that consists of arrows of the same "type". Two arrows have the same type if they either lead from an upper half-flow to an upper half-flow or from a lower to an lower half-flow. *Any chain of same-type arrows can be brought to one of the forms*

$$\underline{vw} \xrightarrow{-\#w} \underline{v} \to \underline{uv}, \tag{48a}$$

$$\overline{uv} \to \overline{v} \xrightarrow{-\#^c w} \overline{vw} \tag{48b}$$

   without loss of strength.

In other words, we can assume that a shortening arrow always occurs before a lengthening one.

To prove this, we first show that if a lengthening arrow is followed by a shortening arrow, they can be rearranged without loss of strength. For lower half-flows, such a chain must have the form (a) $\underline{u} \to \underline{uxv} \xrightarrow{-\#xv} \underline{v}$, or (b) $\underline{uv} \to \underline{uvw} \xrightarrow{-\#w} \underline{vw}$. Form (a) occurs when the first arrow adds cell states that the second takes away, while (b) occurs when there is a common substring v in the first and third half-flow of the chain.

But in (a), we can remove x and get a stronger arrow chain. This chain is the special case of (b) with $v = \epsilon$, so that we only need to consider (b). And (b) can be replaced with (48a), so that we have proved our assertion for chains of two arrows.

Chains of more than two arrows can now be rearranged so that there is first a chain of shortening and then one of lengthening arrows. But arrows of the same form can be condensed to a single arrow, as we have seen before. So we end up again with (48a).

Chains of less than two arrows can be extended by adding "empty" arrows, say by setting $u = \epsilon$. Therefore all sequences of arrows between lower half-flows can be brought into the form (48a). The proof for upper half-flows is similar.

3. Another simplification concerns the *type-changing* arrows in (47c). We can assume that *two type-changing arrows never occur directly in sequence*. For if they occur, as in $\underline{v} \to \overline{v} \xrightarrow{(|v|-\ell)C} \underline{v}$, we can remove them both and get an arrow chain that is at least as strong.

4. Next we consider a chain of three arrows with a type-changing arrow in its centre. As we now can conclude, they can only have the following two forms (with $uv = u'v'$ or $vu = v'u'$, respectively):

$$\underline{v} \to \underline{uv} \to \overline{u'v'} \to \overline{v'}, \tag{49a}$$

$$\overline{v} \xrightarrow{-\#^c u} \overline{vu} \xrightarrow{(|vu|-\ell)C} \underline{v'u'} \xrightarrow{-\#u'} \underline{v'}\,. \tag{49b}$$

But for them we can assume that either $u = \epsilon$ or $u' = \epsilon$ and simplify the arrow chains accordingly. (In other words, *do not take away what you just have added*.)

The proof consists of four cases. For for each arrow chain, one must distinguish between $|u| \geqslant |u'|$ and $|u| \leqslant |u'|$. We will look only at one case, namely that in which $|u| \leqslant |u'|$ is true in (49b). There we can write $u' = xu$ for a suitable $x \in \Sigma^*$,

$$\overline{v} \xrightarrow{-\#^c u} \overline{vu} \xrightarrow{(|vu|-\ell)C} \underline{v'xu} \xrightarrow{-\#xu} \underline{v'}, \tag{50}$$

and then remove u to get a stronger chain,

$$\overline{v} \to \overline{v} \xrightarrow{(|v|-\ell)C} \underline{v'x} \xrightarrow{-\#x} \underline{v'}\,. \tag{51}$$

The other cases are similar and always lead to a new chain that is at least as strong as the original one.

5. Now we can construct the two chains that lead to inequality (42) of the theorem:

$$\underline{a} = \underline{uvw} \xrightarrow{-\#w} \underline{uv} \to \overline{uv} \to \overline{v} \xrightarrow{-\#^c w'} \overline{vw'} \xrightarrow{(|vw'|-\ell)C} \underline{vw'} \to \underline{u'vw'} = \underline{b}, \tag{52a}$$

$$\overline{a} = \overline{uvw} \to \overline{vw} \xrightarrow{(|vw|-\ell)C} \underline{vw} \xrightarrow{-\#w} \underline{v} \to \underline{u'v} \to \overline{u'v} \xrightarrow{-\#^c w'} \overline{u'vw'} = \overline{b}\,. \tag{52b}$$

   We must collect the arrow chains for all possible u, v, w, u', $v' \in \Sigma^*$ to get all requirements on $f(b)$ for a given $f(a)$.

   The chains arise naturally once we note that a has maximal length and that therefore the first arrow must necessarily be shortening. After that, there is only one possible successor for each arrow, until the arrow chain ends in b.

   The first chain has the weight $-\#w - \#^c w' - |u'|C$ and the second, $-\#w - \#^c w' - |u|C$, from which we can see that (42) is the right formula.

6. Our argument is not yet complete. The arrow chains of (52) consist of a single cycle of shortening and lengthening arrows, from uvw to $u'v'w'$. What if we created arrow chains of more than one such cycle? Would we then get more conditions on $f(b)$?

   To resolve this question, we first need shorter expressions for the cycles. We write the chains in (52) as single arrows, as

$$\underline{uvw} \xrightarrow{-|u'|C-\#w-\#^c w'} \underline{u'v'w'}, \tag{53a}$$

$$\overline{uvw} \xrightarrow{-|u|C-\#w-\#^c w'} \overline{u'v'w'}\,. \tag{53b}$$

   The two cycles have essentially the same form, so it will be enough to consider only the first one.

   When we now connect two arrows of the form (53a), the result can always be written as

$$\underline{uxyzw} \xrightarrow{-|u'|C-\#w-\#^c w'} \underline{u'xyzw'} \xrightarrow{-|u''|C-\#zw-\#^c w''} \underline{u''yw''}\,. \tag{54}$$

   This is because the action of each arrow can be understood as taking away cell states from both sides of the string at its left and then adding others, resulting in the string at its right. (In (53), the regions u and w are removed and u' and w' then added.) A region in the centre is left unchanged. With two arrows, the unchanged region of the first arrow might be shortened by the second. In (54) we therefore have assumed, that xyz is the unchanged region of the first and y that of the second arrow.

   The strength of the arrow chain in (54) is $-|u'w'u''|C - \#zw - \#^c w''$. But we can achieve the same result with a single arrow,

$$\underline{uxyzw} \xrightarrow{-|u''|C-\#zw-\#^c w''} \underline{u''yw''}\,. \tag{55}$$

   Its strength differs from that of (54) by $-|u'w'|C$, which is never a positive number. This means that we can replace (54) with (55) in a chain of arrows and, by induction, that a single cycle (53) is enough.

7. Finally, what about shorter chains? One could omit the type-changing arrows and get chains of the form

$$\underline{a} = \underline{vw} \xrightarrow{-\#w} \underline{v} \to \underline{u'v} = \underline{b}, \tag{56a}$$

$$\overline{a} = \overline{uv} \to \overline{v} \xrightarrow{-\#^c w'} \overline{vw'} = \overline{b}, \tag{56b}$$

which look as if they could be stronger that those in (52). But in fact they are just special cases of these chains. One can e. g. see that (56a) is just (52b) with $u = w' = \epsilon$. Recall that $\underline{a} = \overline{a}$ and $\underline{b} = \overline{b}$ and note that, since $|vw| = \ell$, the arrow $\overline{vw} \xrightarrow{(|vw|-\ell)C} \underline{vw}$ in (52b) has strength 0. In the same way one can see that (56b) is a special case of (52a).

We have now shown that all relations between $f(b)$ and $f(a)$ derive from the arrow chains in (52). Therefore the conditions in (42) define $m(a, k)$.[4] □

6.4. Examples

With Theorem 6.2, we can now find examples for minimal number-conserving automata. As before with the elementary cellular automata, we use the minimal state set $\Sigma = \{0, 1\}$. So we have $C = 1$, and every cell may contain at most one particle. Even with these restrictions, we can find cellular automata with an interesting behaviour.

The influence of the particle density An example is $m(0110, 2)$ in Figure 3. In the figure, cells in state 0 are displayed in white, and cells in state 1 in black. Time runs upward, and the line at the bottom is a random initial configuration.

In an ordinary cellular automaton, one can expect that the number of ones and zeroes in a random initial configuration has no great influence on the patterns that arise after a few generation. Under most rules, the fraction of cells in state 1 changes over time. With number-conserving automata this is no longer true. Figure 3 therefore contains four evolutions of $m(0110, 2)$ with different densities: the density is the fraction of cells in state 1 in the initial configuration – which then stays constant during the evolution of the cellular automaton.

Especially in the two low density evolutions (with density $= 0.1$ and 0.2), one can see that particles move with three possible speeds: Isolated particles move with speed 1, blocks of two particles, like 0110, move with speed 3, and blocks of three particles move 5 steps over two generations (from 0111000000 via 0001101000 to 0000001110) and therefore have a speed of 2.5. For very low densities like 0.1, the evolution is initially dominated by non-interacting single particles, but we can already see how they are collected by the faster structures and integrated into their particle stream. In the density 0.2 image we can see how the fast structures interact: When a middle speed particle group interacts with a fast one, a short "traffic jam" of high particle density occurs, but then the two particle groups

[4] This proof is a bit long. Another way to prove this theorem – possibly shorter but less natural – is to derive the inequalities (42) only as necessary conditions and then to verify that the function f defined by them satisfies $f(a) \geqslant k$ and the flow conditions. If the conditions (42) were not sufficient, f would be smaller than all the flows g with $g(a) \geqslant k$ and therefore not be a flow.

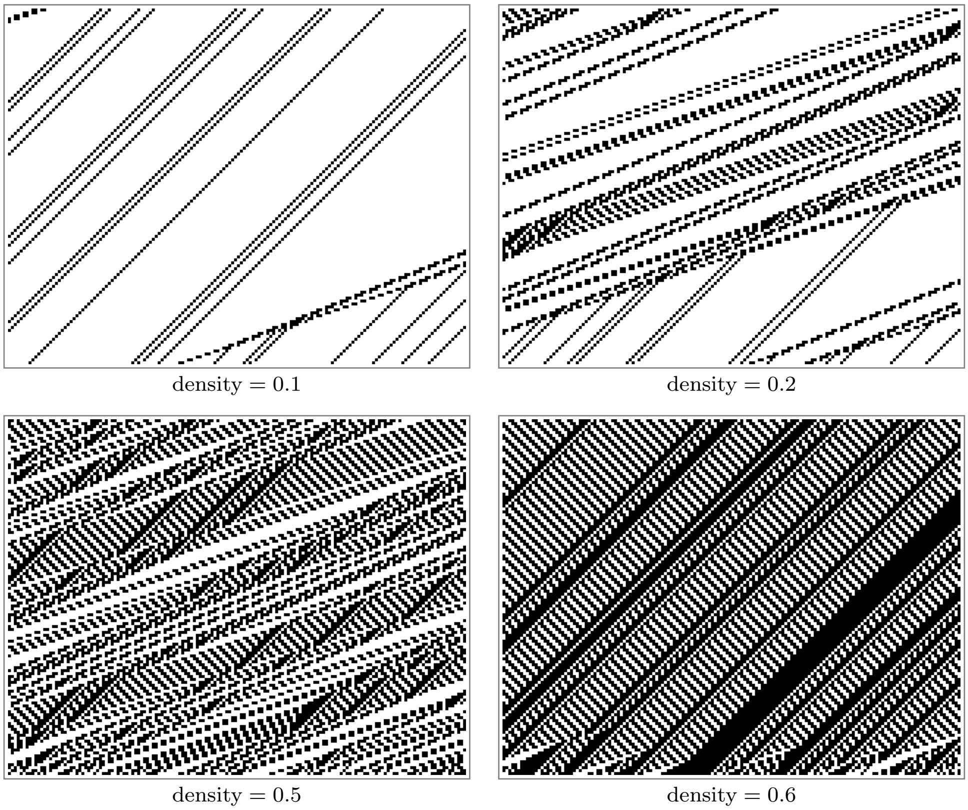


Figure 3. Rule $m(0110, 2)$ at different densities. Time goes upward.

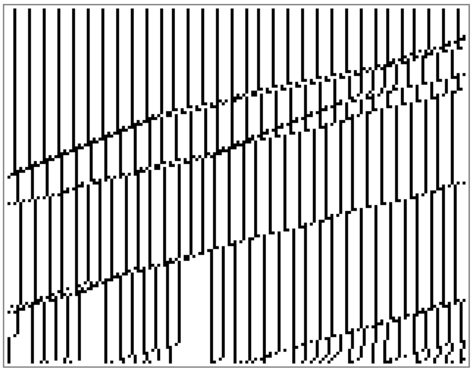
separate again. (Note that the middle-speed groups before and after the interaction consist of different particles.) With density 0.5, the traffic jams are more common, and also highly regular structures between them. With the moderately high density of 0.6, all interesting behaviour stops early, and the automaton looks like σ^1.

So $m(0110, 2)$ already establishes a vaguely traffic-like behaviour, except that the speed of a particle also depends on the location of the particles next to it.

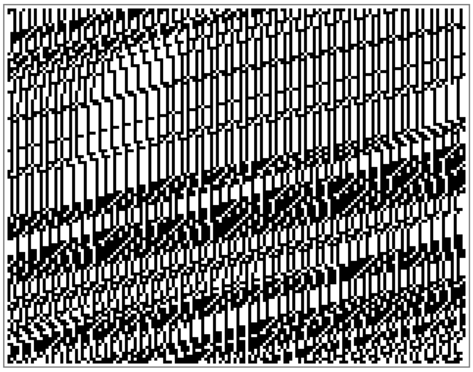
Other phenomena Figure 4 illustrates other phenomena that occur with minimal number-conserving cellular automata.

Often, especially at low densities, nothing or almost nothing happens. The evolution of rule $m(11010, 2)$ at the top left of the figure is an example. Most of the time, almost all particles are arranged in a pattern in that no non-zero flow arises. From time to time, a disturbance moves over the cells.

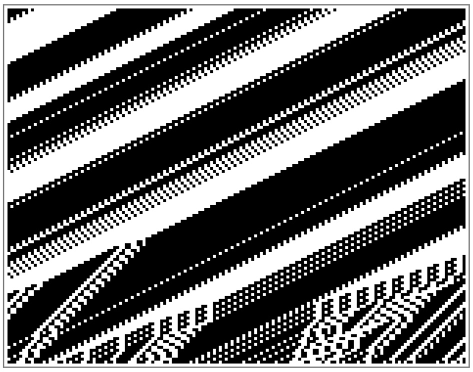
Rule $m(01110, 2)$, at the top right, consists too of times of inactivity and moving high-density regions. The high-density regions have however their own intricate structure.



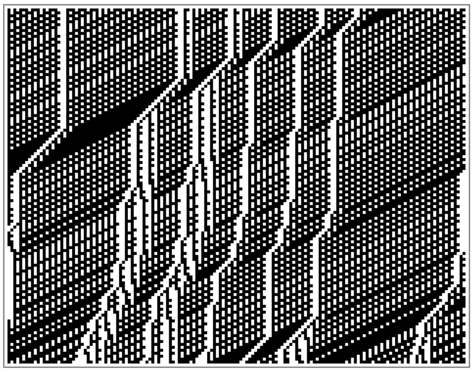
$m(11010, 2)$ at density 0.25



$m(01110, 2)$ at density 0.55



$m(01100, 2)$ at density 0.65



$m(01100, 1)$ at density 0.65

Figure 4. Some minimal rules.

The evolution of rule $m(01100, 2)$, at the bottom left, has large regions of the highest possible density, and the disturbances in them sometimes look like “anti-particles”: periodic structures of emptiness between particles.

At the bottom right, the related rule $m(01100, 1)$ shows a different pattern of almost regular high-density regions. The disturbances in them show a complex, tree-like pattern.

7. Odds and ends

7.1. Non-deterministic number conservation

So far, the possibility that two or more states have the same particle content has rarely been addressed. Such non-minimal state sets have however an important application: They enable the construction of non-deterministic number-conserving cellular automata.

In a non-deterministic cellular automaton, the value of the transition function φ is a set of states, and the next state of a cell may be any element of the value of φ, when applied to its neighbourhood. Instead of (6), we have

$$\hat{\varphi}(a)_x \in \varphi(a_{x-r_1}, \dots a_{x+r_2}) \quad \text{for all } x \in \mathbb{Z}. \tag{57}$$

When we then have constructed a flow function f for a non-minimal state set, we can construct a non-deterministic number-conserving rule with the help of a relaxed form of (13), namely

$$\varphi(w) \subseteq \{\, \alpha \in \Sigma\colon \#\alpha = f(w_{0:\ell}) + \#w_{r_1} - f(w_{1:\ell}) \,\}\,. \tag{58}$$

Such a function is clearly number-conserving, since all possible choices for the next state of a cell have the same particle content.

The condition that for each $w \in \Sigma^{\ell+1}$, all elements of $\varphi(w)$ must have the same particle content is also necessary: If there are $\alpha, \beta \in \varphi(w)$ with $\#\alpha \neq \#\beta$, we can use a configuration a which contains w as a substring to construct a counterexample. Among the possible successor configurations of a in the next time step, there must be two that only differ at one cell, which is in one configuration in state α and in the other one in state β. Since $\#\alpha \neq \#\beta$, number conservation cannot hold for both configurations.

7.2. Two-sided neighbourhoods

We now return to the case where r_1 and r_2 are arbitrary non-negative numbers. The flow functions for such *two-sided neighbourhoods* are related in a very simple way to the one-sided neighbourhoods that we have so far investigated.

Let $\hat{\varphi}$ be a global transition rule with arbitrary r_1 and r_2 and let $\hat{\varphi}'$ be the transition rule with a one-sided neighbourhood that is related to it. Let f and f' be their flow functions. Then we can write

$$\hat{\varphi} = \hat{\varphi}' \circ \hat{\sigma}^{-r_2}\,. \tag{59}$$

This is because we can get the effect of $\hat{\varphi}$ by first moving the content of all cells by r_2 positions to the left and then applying $\hat{\varphi}'$. The corresponding equation for flows is

$$f(uv) = f'(uv) - \#v, \tag{60}$$

for all $u \in \Sigma^{r_1}$ and $v \in \Sigma^{r_2}$: The shift $\hat{\sigma}^{-r_2}$ produces an additional flow of $-\#v$ particles over the boundary between u and v.

As a corollary of (60), we see that the new two-sided flows obey the same partial order as the one-sided do. If g is another flow with radii r_1 and r_2 and f' is its one-sided equivalent, then

$$f \leqslant g \qquad \text{iff} \qquad f' \leqslant g', \tag{61}$$

as we easily can conclude from (60) and the definition of the partial order in (33).

The theory of minimal flows for two-sided neighbourhoods is therefore isomorphic to that for one-sided neighbourhoods. As a good notation for two-sided minimal flows I would propose

$$m(u, v; k) = m(uv, k) - \#v\,. \tag{62}$$

8. Open questions

The set of all number-conserving one-dimensional cellular automata has, as we have seen, an intricate structure. It leads to many open questions, of which I will list a few, with comments:

1. How many number-conserving automata are there for a given state set Σ and flow length ℓ?

2. How many flow functions are there for given Σ and ℓ?

   The answers to this and the previous question are the same if Σ is a minimal state set. For both questions, an explicit formula as answer is probably very complex. An asymptotic formula could be easier to find and might provide more insight.

   Boccara and Fukś [6] have already found that for $\Sigma = \{0, 1\}$, there are 5 rules for $\ell = 2$, 22 rules for $\ell = 3$ and 428 for $\ell = 4$.

3. If two number-conserving automata have the same flow function, how is their behaviour related?

   An answer to this question would provide insight into the relation between cellular automata and their flow functions. It would also be helpful for the better understanding of non-deterministic number-conserving automata.

4. If the behaviour of the automata with flow functions f and g is known, what can be said about those with flows $f \wedge g$ and $f \vee g$?

   Ideally, the lattice structure of the flows would provide information about the cellular automata. This problem should first be investigated for minimal state sets, since otherwise it would require an answer to the previous question.

5. Given $\underline{f}_k$ and $\overline{f}_k$, what can be said about $\hat{\varphi}$?

   The pairs $(\underline{f}_k, \overline{f}_k)$ provide a classification of number-conserving cellular automata, and we should expect that they group automata with similar behaviour. But what does "similar" mean in this context?

   An argument similar to that for Figure 1 gives a partial result: $\overline{f}_0 = \ell C$ enforces that $\hat{\varphi}$ is the shift function $\hat{\sigma}^\ell$.

6. What kind of lattices are the flow sets $\mathcal{F}(\ell, \Sigma)$?

   We can see in Figure 2 that $\mathcal{F}(2, \{0, 1\})$ is a linear order. But there are larger values of ℓ for which $\mathcal{F}(\ell, \{0, 1\})$ does contain incomparable elements. One example is shown in Figure 5 and 6.

7. What does the theory for more than one kind of particle look like?

   There are two types of multi-particle automata. In automata of the first type, each cell contains several containers, each for one kind of particle, and the particles only move between "their" containers. This kind of automaton has a theory that is a straightforward extension of the one-particle theory.

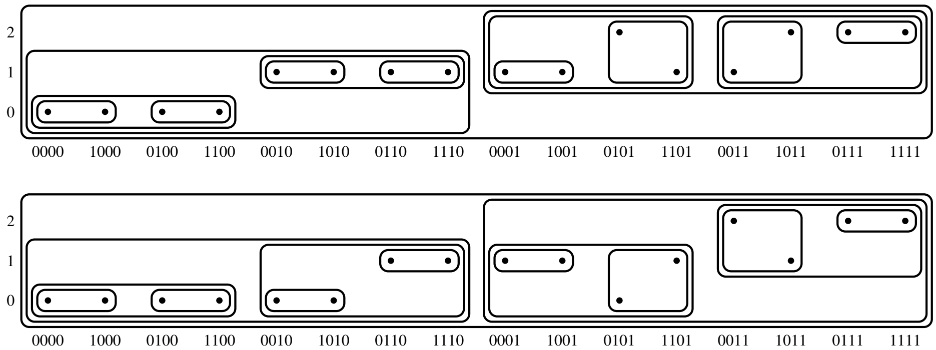
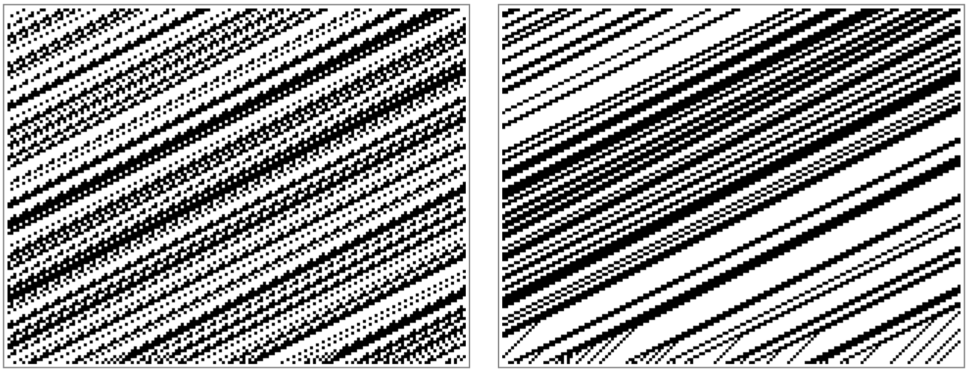


Figure 5. Two incomparable flows, $m(0101, 2)$ and $m(0011, 2)$.



$m(0101, 2)$ $m(0011, 2)$

Figure 6. The flows $m(0101, 2)$ and $m(0011, 2)$ at density 0.4.

In automata of the second type, there is only one container in each cell and particles of different "colours" that move between the the cells. Here the theory will be more complex.

8. What about particles in higher dimensions?

   The derivation of Theorem 4.3 only works in one dimension. For higher-dimensional cellular automata therefore new ideas are needed.

9. Are there practical or theoretical applications for this theory?

   With practical applications I mean e. g. simulations of physical systems. A theoretical application could be the construction of a universal number-conserving cellular automaton or something similar.

10. Can the theory be simplified?

    In the proofs and calculations of this paper, a small number of types of inequalities are used over and over again. Is there a theory with which the repetitions can be compressed into a few lemmas at the beginning, such that the actual proofs take only (say) two pages? This would also be helpful for multi-particle and higher-dimensional systems.

Acknowledgements Barbara Wolnik provided helpful comments on an earlier version of this article. A very thorough anonymous reviewer also contributed much to the quality of the text.

References

[1] Hattori T, Takesue S. Additive conserved quantities in discrete-time lattice dynamical systems. *Physica D*, 1991. **49**(3):295–322. doi:10.1016/0167-2789(91)90150-8.

[2] Pivato M. Conservation Laws in Cellular Automata. *Nonlinearity*, 2002. **6**:1781. doi:10.1088/0951-7715/15/6/305.

[3] Imai K, Alhazov A. On Universality of Radius 1/2 Number-Conserving Cellular Automata. In: Calude CS, Hagiya M, Morita K, Rozenberg G, Timmis J (eds.), Unconventional Computation. Springer Verlag, Berlin, Heidelberg. 2010 pp. 45–55. ISBN-978-3-642-13523-1.

[4] Bhattacharjee K, Naskar N, Roy S, Das S. A Survey of Cellular Automata: Types, Dynamics, Non-uniformity and Applications. *Natural Computing*, 2016. pp. 1–29. doi:10.1007/s11047-018-9696-8.

[5] Wolnik B, Nenca A, Baetens JM, De Baets B. A split-and-perturb decomposition of number-conserving cellular automata. *Physica D*, 2020. **413**:132645. doi:10.1016/j.physd.2020.132645.

[6] Boccara N, Fukś H. Cellular automaton rules conserving the number of active sites. *Journal of Physics A*, 1998. **31**:6007–6018. doi:10.1088/0305-4470/31/28/014.

[7] Fukś H. A class of cellular automata equivalent to deterministic particle systems. In: Feng S, Lawniczak AT, Varadhan SRS (eds.), Hydrodynamic Limits and Related Topics. AMS, Providence, RI, 2000 pp. 57–69.

[8] Durand B, Formenti E, Róka Z. Number-conserving cellular automata I: decidability. *Theoretical Computer Science*, 2003. **299**(1-3):523–535. doi:10.1016/S0304-3975(02)00534-0.

[9] Wolfram S. Statistical Mechanics of Cellular Automata. *Reviews of Modern Physics*, 1983. **55**:601–644. doi:10.1103/RevModPhys.55.601.

[10] Davey BA, Priestley HA. Introduction to Lattices and Order. Cambridge University Press, Second edition, 2002. ISBN-13:978-0521784511, 10:0521784514.