

Surprisingly complex number-conserving binary Moore neighborhood cellular automata

Dominik Falkiewicz¹, Barbara Wolnik^{2,4}, Witold Bołt³, Adam Rutkowski¹, and Bernard de Baets⁴

¹ Institute of Theoretical Physics and Astrophysics, Faculty of Mathematics, Physics and Informatics, University of Gdańsk, Wita Stwosza 57, 80-308 Gdańsk, Poland

² Institute of Mathematics, Faculty of Mathematics, Physics and Informatics, University of Gdańsk, Wita Stwosza 57, 80-308 Gdańsk, Poland

³ Jit Team, Łużycka 8C, 81-537 Gdynia, Poland

⁴ KERMIT, Department of Data Analysis and Mathematical Modelling, Faculty of Bioscience Engineering, Ghent University, Coupure links 653, B-9000 Gent, Belgium

Abstract. Until recently, only a few number-conserving binary Moore neighborhood cellular automata had been known. Last year we presented the (Ω, A) framework that allows one to generate hundreds of thousands of such rules, with the additional advantage of providing a simple and consistent way to interpret changes in the cellular space as movements of individually identifiable ones that could represent physical particles. Since then, using a recursive algorithm, we managed to generate all number-conserving binary 2-dimensional Moore neighborhood cellular automata. In this paper, we present examples of number-conserving rules that fall outside of the (Ω, A) framework and exhibit surprisingly complex behavior.

Keywords: Cellular automata · number conservation · Moore neighborhood.

1 Introduction

A d -dimensional cellular automaton (CA) is a quadruple $(\mathcal{C}, Q, V, f)$, where

- $\mathcal{C} = \mathbb{Z}^d$ is the cellular space;
- Q is the finite set of states;
- $V = (\vec{v}_1, \vec{v}_2, \dots, \vec{v}_m)$ is the neighborhood vector, which means that, for a cell $\mathbf{i} \in \mathcal{C}$, the cells $\mathbf{i} + \vec{v}_1, \mathbf{i} + \vec{v}_2, \dots, \mathbf{i} + \vec{v}_m$ are its neighbors;
- $f : Q^m \rightarrow Q$ is the local rule: the state of a cell $\mathbf{i}$ in the next time step is given by $f(x_1, x_2, \dots, x_m)$, where $x_1, x_2, \dots, x_m$ are the current states of the cells $\mathbf{i} + \vec{v}_1, \mathbf{i} + \vec{v}_2, \dots, \mathbf{i} + \vec{v}_m$, respectively.

A *configuration* is any mapping from the grid $\mathcal{C}$ to Q ; the set of all configurations is denoted by X .

A given local rule f induces a *global rule* $F : X \rightarrow X$ defined for any $\mathbf{x} \in X$ and $\mathbf{i} \in \mathcal{C}$ as follows

$$F(\mathbf{x})_{\mathbf{i}} = f(x_{\mathbf{i}+\vec{v}_1}, x_{\mathbf{i}+\vec{v}_2}, \dots, x_{\mathbf{i}+\vec{v}_m}).$$

Usually, the global rule F is associated with the cellular automaton.

Definition 1. *A CA with the global rule F is number-conserving if for every finite configuration $\mathbf{x} \in X$ it holds that $\sum_{\mathbf{i} \in \mathcal{C}} F(\mathbf{x})_{\mathbf{i}} = \sum_{\mathbf{i} \in \mathcal{C}} \mathbf{x}_{\mathbf{i}}$.*

For simplicity, we will say that a local rule is number-conserving if the corresponding global rule is number-conserving.

Throughout this study, we will only deal with binary CAs (*i.e.*, $Q = \{0, 1\}$).

In d -dimensional space, the Moore neighborhood is defined by the Chebyshev distance

$$\text{dist}_{\text{Ch}}(\mathbf{i}, \mathbf{j}) = \max_{1 \leq k \leq d} |i_k - j_k|, \quad (1)$$

in the following way: the neighborhood of a cell $\mathbf{i}$ is a closed unit ball centered at $\mathbf{i}$, *i.e.*, consisting of those cells whose Chebyshev distance to $\mathbf{i}$ does not exceed 1. For example, in the case of two dimensions, the Moore neighborhood is given by

$$V_M = (-\vec{v}_4, \vec{v}_2, \vec{v}_3, -\vec{v}_1, \vec{v}_0, \vec{v}_1, -\vec{v}_3, -\vec{v}_2, \vec{v}_4),$$

where $\vec{v}_0 = [0, 0]$, $\vec{v}_1 = [1, 0]$, $\vec{v}_2 = [0, 1]$, $\vec{v}_3 = [1, 1]$, and $\vec{v}_4 = [1, -1]$.

Although the necessary and sufficient conditions for a CA to be number-conserving presented in [1], [2] can be formulated for any dimension, until now they have never been used to find a complete list of number-conserving CAs for any dimension $d \geq 2$ and any set of states, due to their complexity. For this reason, if $d \geq 2$ little is known about number-conserving d -dimensional cellular automata with the Moore neighborhood (even for the binary case).

In the remainder of the paper, we will consider only Moore neighborhood 2-dimensional number-conserving binary CAs. We will refer to them as NCCAs.

2 The set of all NCCAs

As already mentioned, until recently only a few NCCAs had been known. In particular, the identity, shifts and simple traffics are number conserving [4]. All 74 NCCAs have been found for 2×3 -neighborhood (a reduced Moore neighborhood without the top row of neighbors) [3], which can be easily translated (by rotating them) into 296 NCCAs in the Moore neighborhood. In our recent work [5], we proposed a framework for the description and construction of NCCAs. Instead of defining the LUT of an automaton and then inferring its behavior, we described a certain class of NCCAs in terms of their behavior, that can be compared to movement of uniform particles represented by ones in the cellular space. This behavior can be understood as the composition of a global rule, which we denote with Ω and a set of local traffic-like transformations, which we denote with Λ . We have shown that in the case of 2×3 -neighborhood, this class represents all

number-conserving binary rules [5]. Our hope was that this would also be the case for the full Moore neighborhood; however, as we will see, it is not.

Using the necessary and sufficient condition [2] we designed a recursive algorithm that allowed us to find all NCCAs for the Moore neighborhood, which was thought impossible due to the complexity of searching through 2^{512} local rules. The complete list of 147 309 433 NCCAs can be found here:

https://drive.google.com/file/d/1qq0jBEfSYCfkgKB-kMKxY_uy24yIEVOT6/view?usp=sharing

The file contains hexadecimal representation of the NCCAs. In this paper, we will not be focusing on the algorithm, but rather we will present a few of the more interesting NCCAs that we so far identified and the method we used to look for them within the whole set.

Let us set the following requirements for the local rule. A local rule guaranties that:

1. The middle section of any horizontal line surrounded by zeroes will be stable.
2. At the right end of any horizontal line of ones surrounded by zeros, the final one is removed in each time step.
3. A one is added at the left end of any horizontal line of ones surrounded by zeros in each time step.
4. Any vertical line of ones with two columns of zeroes on its left will not have additional ones appearing directly to the left of it.

If such a rule existed, it could be number-conserving, but it could not be described using the (Ω, Λ) framework. Requirements 1 – 3 exclude rules where Ω is not shift-left and requirement 4 excludes rules where Ω is shift-left. It is quite easy to design a regular expression that will match only the hexadecimal codes of rules that fulfill requirements 1 – 4. First, we translate the requirements into LUT-like form, where f represents the local rule:

1. Requirement: $f \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} = 0$, $f \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} = 1$, $f \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = 0$
2. Requirement: $f \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} = 0$
3. Requirement: $f \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} = 1$
4. Requirement: $f \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} = 0$, $f \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} = 0$, $f \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} = 0$

We use hexadecimal representation of LUTs to save space and make the data more legible. For requirements 3 and 4 to be kept, the 9th bit of the LUT from the right must be 1, and the 10th bit must be 0. After translating the LUT

into the hexadecimal form, these bits will be the lowest value bits of the 3rd hexadecimal digit from the right, which can therefore only take the values of 1, 5, 9 or D. Using this method for all of the LUT requirements, we construct the following regular expression:

`.{15}[02468ACE].{93}[048C].{3}[13579BDF].[02468ACE].{9}[159D][01234567].`

There are 362 854 local rules that match this expression. Because of the behavior these local rules exhibit owing to requirement 4, we call them "Wall Rules". We only had a brief look at a few of these, but among them, we already found rules exhibiting completely different and very complex dynamics. Two of these NCCAs are presented below.

3 Examples of interesting NCCAs

3.1 Wall Rule 1

The Wall Rule 1 is given by the following hexadecimal code:

```
FFFFCC40 CF0FCC40 FFFFEE40 CF0FEE40
FFFF8840 C3038840 FF00DD40 C303DD40
0000CC40 FF3FCC40 FFFFEE40 FF3FEE40
FFFF8840 CC0C8840 FF00DD40 CC0CDD40
```

Starting from an initial configuration that is randomly generated with uniform probabilities for ones and zeros, this rule tends to quickly collect ones into irregular structures. These are then rearranged by streams of ones that follow a traffic-like movement pattern along the right edge of the structures, and sets of lines of ones arranged along bottom-left to top-right axis and traveling from bottom-right to top-left that "shave" columns of ones from the structures on their top-left end.

Example evolutions of Wall Rule 1 are presented in Figure 1 and Figure 2.

3.2 Wall Rule 2

The rule is given by the following hexadecimal code:

```
F7FFFFC8 FF407F08 F7FFFFC8 00407F08
F7FFCCC8 00404C08 F70099C8 00401908
F7FFFFC8 0F4F7F08 F700FFC8 0F4F7F08
F700CCC8 3C7C4C08 F70099C8 3C7C1908
```

Starting from an initial configuration that is randomly generated with uniform probabilities for ones and zeros, Wall Rule 2 quickly rearranges the ones into

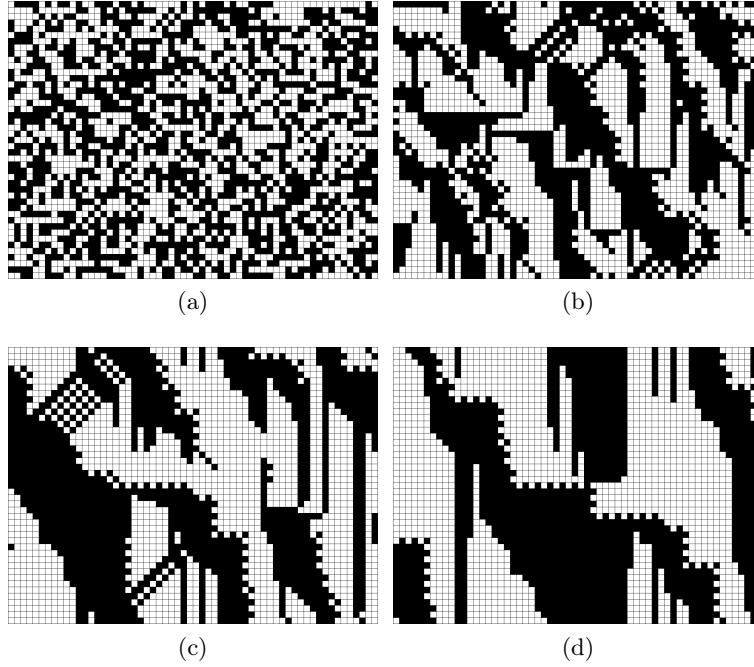


Fig. 1: The initial grid (a) organizes itself into small structures (b) that then combine (c). We observe diagonal sets of lines moving from bottom-left to top-right. They reshape other structures, usually shaving multiple columns of ones from the structure on their top-right side and adding these ones to the structure on their bottom-left side. sometimes they disappear (d), which can results in cellular space-spanning cycle where the intermittent sequence of ones travels along the surface of a single structure on the torus.

fluid configurations that resemble flames. These evolve seemingly completely chaotically, sometimes forming dense sections of ones, that on the left side end with straight, vertical border. If this border extends to span the entire torus, the structure stabilizes, and eventually most ones are gathered into a single dense band. However, most of the time the structure dissolves back into a flame-like evolving pattern. It is not immediately obvious what governs these phase changes.

An example of the evolution of a cellular space for this rule is presented in Figure 3.

3.3 What next?

It is important to point out that the rules presented here are not carefully selected to be the most impressive. They are just some of the rules we happened to look

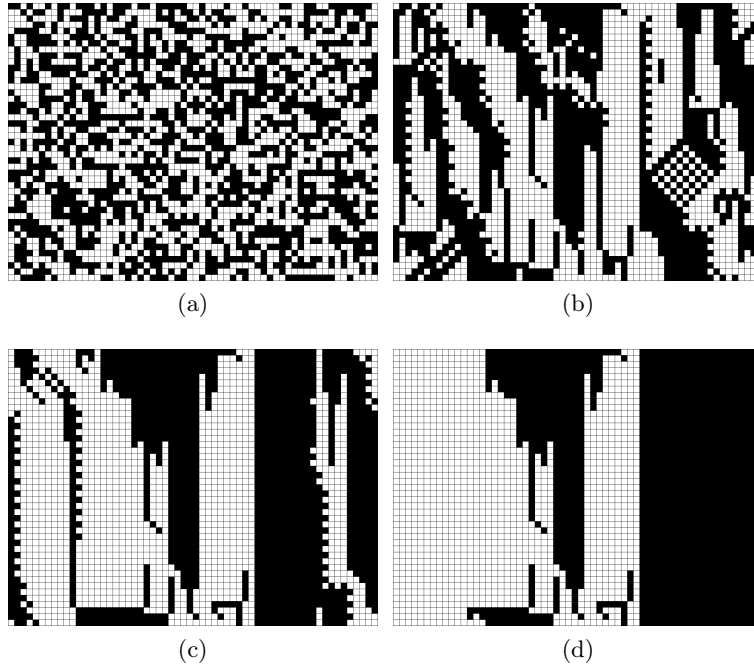


Fig. 2: The initial grid (a) organizes itself into small structures (b) that then combine, with a single, vertical line of ones spanning the torus (c). When that happens, most ones are usually collected into a single band, ending in a column of intermittent ones traveling in a traffic from bottom to top (d). In this case there is also a complex structure left that is not part of this this main band.

at. Many other rules are expected to behave very similarly. Many more are expected to exhibit behaviors that are just as complex but completely different from those demonstrated here. The diversity and complexity of the number-conserving binary 2-dimensional Moore neighborhood cellular automata defied all of our expectations. Full explanation of the dynamics of any one of them can pose a similar challenge to the description of the famous Conway’s Game of Life, which after all is also just one binary rule.

References

- [1] T. Hattori and S. Takesue, “Additive conserved quantities in discrete-time lattice dynamical systems,” *Physica D: Nonlinear Phenomena*, vol. 49, no. 3, pp. 295–322, 1991, ISSN: 0167-2789. DOI: 10.1016/0167-2789(91)90150-8.
- [2] B. Durand, E. Formenti, and Z. Róka, “Number-conserving cellular automata i: Decidability,” *Theoretical Computer Science*, vol. 299, no. 1, pp. 523–535, 2003, ISSN: 0304-3975. DOI: 10.1016/S0304-3975(02)00534-0.

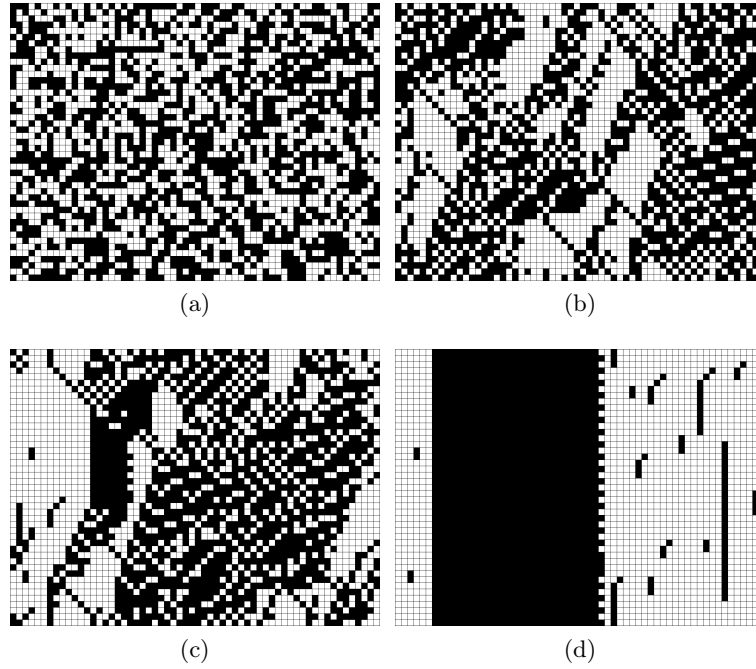


Fig. 3: The initial grid (a) organizes itself into highly mobile structures that resemble waves or flames (b) with ones traveling from right to left. sometimes, seemingly randomly a continuous band of ones preceded by a band of zeroes begins to form (c). It often looks like one is about to form, but then the dense pack of ones dissolves back into a flame-like pattern. However, after such a band forms it is stable, and it gathers most of the remaining ones except for some smaller, stable structures.

- [3] H. Ishizaka, Y. Takemura, and K. Imai, “On enumeration of motion representable two-dimensional two-state number-conserving cellular automata,” in *2015 Third International Symposium on Computing and Networking (CANDAR)*, 2015, pp. 412–417. DOI: 10.1109/CANDAR.2015.52.
- [4] N. KADACHE and R. SEGHIR, “A symmetric identity-rule-variation-based method for enumerating and building radius-1 two-state ncca rules,” *Advances in Complex Systems*, vol. 27, no. 06, p. 2450010, 2024. DOI: 10.1142/S0219525924500103. eprint: <https://doi.org/10.1142/S0219525924500103>. [Online]. Available: <https://doi.org/10.1142/S0219525924500103>.
- [5] B. Wolnik, D. M. Falkiewicz, W. Bołt, A. Rutkowski, and B. De Beats, “Conditional traffic-like rules for particle-flow simulation in cellular automata,” *arXiv*, 2025. DOI: <https://doi.org/10.48550/arXiv.2512.08727>.