Physics Letters A 376 (2012) 2645–2657

Contents lists available at SciVerse ScienceDirect

Physics Letters A

www.elsevier.com/locate/pla

Universal map for cellular automata

V. García-Morales *

Institute for Advanced Study – Technische Universität München, Lichtenbergstr. 2a, D-85748 Garching, Germany

ARTICLE INFO

Article history:
Received 24 April 2012
Received in revised form 2 July 2012
Accepted 19 July 2012
Available online 23 July 2012
Communicated by C.R. Doering

Keywords:
Theory of computation
Mathematical results
Complexity
Dynamics

ABSTRACT

A universal map is derived for all deterministic 1D cellular automata (CAs) containing no freely adjustable parameters and valid for any alphabet size and any neighborhood range (including non-symmetrical neighborhoods). The map can be extended to an arbitrary number of dimensions and topologies and to arbitrary order in time. Specific CA maps for the famous Conway's Game of Life and Wolfram's 256 elementary CAs are given. An induction method for CAs, based in the universal map, allows mathematical expressions for the orbits of a wide variety of elementary CAs to be systematically derived.

1. Introduction

Cellular automata (CAs) constitute paradigmatic models of complexity in nature [1–3], from snowflakes, patterns in mollusc seashells and spiral waves in the Belousov–Zhabotinsky reaction to neural networks and the fundamental physical reality [4]. CAs serve as models for complex natural systems made of large numbers of identical parts [1–7]. They consist of a discrete lattice of sites, with a finite set of possible values each. The site values evolve synchronously in discrete time steps according to identical rules, being determined by the previous values on the sites of their neighborhood. The concept of cellular automaton (CA) dates back over half a century to the efforts of John von Neumann to design self-replicating artificial systems capable of universal computation [8] (see also [9]). It was however in the early eighties, when Wolfram published a series of groundbreaking papers on the subject [5–7,10,11], that CAs received wide attention from the scientific community. Despite some monographs and further important work [12–21] and tantalizing hints [14,22,23], no theory provided mathematical expressions neither for the CA maps nor for the CA orbits in phase space (when possible) valid for arbitrary neighborhood sizes, number of site values (i.e. arbitrary alphabet size) and dimensions. Mathematical tools for these discrete systems with the practical value that partial differential equations (PDEs) have for continuum systems [17] were absent. This has been a major drawback precluding significant progress in the theoretical modeling of complex systems: although quite a tour de force, no general conclusions could be sharply drawn other than the computational ones [1,12,15]. It has indeed long been thought that no simple mathematical expressions could be given for most CAs and a strong emphasis has been put on computational aspects instead of standard mathematics [1]. This Letter tackles this problem providing the missing mathematical tool for CAs. A universal map encompassing all 1D deterministic CAs is presented here. The map can be easily extended to an arbitrary number of dimensions and topologies. Examples of such extensions are also provided here.

The universal map expresses the spatiotemporal dynamics of any 1D CA in a mathematical form, such that it is clearly exposed how the update rule depends on all parameters governing the CA dynamics: neighborhood size and topology, number of symbols in the alphabet and possible outputs of the CA rule. The map provides the local dynamics for a given site i at time t in an analogous way as a PDE does in continuum systems. Thus, as shown in Section 4, the universal map is not a mere reformulation of the update rule: it is a powerful tool to derive systematically mathematical expressions for the orbits of CA rules (as a function of an arbitrary initial condition) in case they exist. To find such solutions, the *induction method* proves to be – in its generality, when dealing with discrete systems – analogous to *integration* in continuum systems. The induction method for CAs (presented in Section 4.1) is explicitly based on the universal map and can be satisfactorily used, for example, to directly obtain the orbits for all rules involving only two symbols in the alphabet and one or two sites. In the case of Wolfram rules (with three sites) there are some few rules, like Wolfram's rule 110, that are computationally

* Tel.: +49 89 289 10595.
E-mail address: vmorales@ph.tum.de.

0375-9601/$ – see front matter
http://dx.doi.org/10.1016/j.physleta.2012.07.021

irreducible (see Section 4.2) and for which to find such a closed expression for the orbit proves to be an impossible task. Still, the CA map may give much valuable information and insight in such irreducible CA automata rules, yielding an invaluable *qualitative* understanding of the spatiotemporal patterns that arise under CA evolution. (An analogous situation is also encountered in continuum dynamics, for example, in the turbulent regime of the complex Ginzburg–Landau equation [24]: in such a regime one cannot give a closed mathematical solution of the PDE but the *bifurcation* to such behavior can be determined and understood from the PDE.) Furthermore, the universal CA map can reproduce any PDE and any real map (continuum limit) when the number of symbols $p \to \infty$ (in Section 4.3 this is briefly illustrated with the logistic map).

2. Mathematics of rules

A mathematical function can be thought as a rule such that, given a numerical input, the output is read from a table containing a series of configurations to which the input compares. The complete table gives the rule, i.e. what to do with an appropriate input. Usually, the configurations on a table must be sharply and unambiguously distinguished from each other within a certain tolerance. Then the table provides the output corresponding to the configuration that successfully matches the input. It is then said that the output is the result of the rule acting on the input.

The notion of table indicates how a general function of a general argument (for example a real number) can be *quantized* in terms of a certain set of countable (discrete) disjunctive sets (the configurations) that arise at a certain precision or tolerance (the distance separating the closest configurations, which gives the measure μ of the configuration) and which splice together to cover the whole range of values of the general argument. As shown below this idea is related to (but not restricted to) the concept of *simple* function, which plays a central role in the theory of measure and Lebesgue integration [25].

The above paragraphs can be seen as a computational way of thinking Mathematics. The most elementary mathematical structure that imitates computation as described above can now be sought. Each computational process is considered here as an "experiment" and standard mathematics is used to theoretically model such an experiment and to give a mathematical expression for any such an elementary step. Proceeding in this way is advantageous since a general mathematical theory for rule-based dynamical systems, in general, and CAs, in particular, can then be rigorously established.

The most simple rule is now considered, i.e. one containing just only one configuration with one nonzero output. When *input* coincides with *configuration* within a certain *tolerance*, the rule returns *output*. Otherwise, the result is zero. The simplest mathematical structure where all actors involved in the above computational process enter can be formally depicted in the following way:

$$output \times \mathcal{B}(configuration - input,\ tolerance) \tag{1}$$

The function $\mathcal{B}$ has two arguments and, in order for the latter expression to return *output* or zero, it is clear that $\mathcal{B}$ should return 1 or 0, respectively. Specifically, it should return 1 if $|configuration - input| < tolerance$ and 0 otherwise. It is clear then that $\mathcal{B}$ is an even function of its first argument. The mathematical function $\mathcal{B}(x,\epsilon)$ having all these properties is the boxcar function plotted in Fig. 1 and introduced in Definition 1 below.

Definition 1. The boxcar function $\mathcal{B}(x,\epsilon)$ is defined, for ϵ and x real, as

$$\mathcal{B}(x,\epsilon) = \frac{1}{2}\left(\frac{x+\epsilon}{|x+\epsilon|} - \frac{x-\epsilon}{|x-\epsilon|}\right) \tag{2}$$

where $|x|$ is the absolute value function.

The boxcar function is defined over the real numbers and has a discontinuity at $x = \pm|\epsilon|$. When $\epsilon = 1/2$, the boxcar function is usually called a rectangular function. The following definitions of $\mathcal{B}(x,\epsilon)$ are equivalent to Definition 1

$$\mathcal{B}(x,\epsilon) = \frac{1}{2}\big[\operatorname{sgn}(x+\epsilon) - \operatorname{sgn}(x-\epsilon)\big] \tag{3}$$

$$= H(x+\epsilon) - H(x-\epsilon) \tag{4}$$

$$= H\left(1 - \frac{x^2}{\epsilon^2}\right) \tag{5}$$

$$= \chi_{(-|\epsilon|,|\epsilon|)}(x) \tag{6}$$

where $\operatorname{sgn}(x)$ is the sign function, $H(x)$ is the Heaviside step function, and $\chi_{(-|\epsilon|,|\epsilon|)}(x)$ is the indicator function for the interval $(-|\epsilon|, |\epsilon|)$. This last definition in terms of the indicator function makes clear that the tolerance ϵ is directly related to the measure μ of the set where the input argument x falls [25]. The value of *configuration* is the value of x at the center of this set, used as its label. All kinds of mathematical structures that can be constructed through addition and multiplication of simple structures of the form Eq. (1) belongs to what can be termed as $\mathcal{B}$-calculus.

When one has a rule with several different configurations indexed by n so that to $configuration_n$ corresponds an $output_n$, the rule is completely given by summing over all elementary structures describing the action of the rule on each separate $configuration_n$ as given in Eq. (1). Each configuration is *exclusive*, i.e., the input cannot simultaneously be equal to two different $configuration_n$ within *tolerance*. The *output* of the rule is then

$$output = \sum_{n \in table} output_n \times \mathcal{B}(configuration_n - input,\ tolerance) \tag{7}$$

The *tolerance* is related to the distance separating two adjacent configurations on the table and is defined as

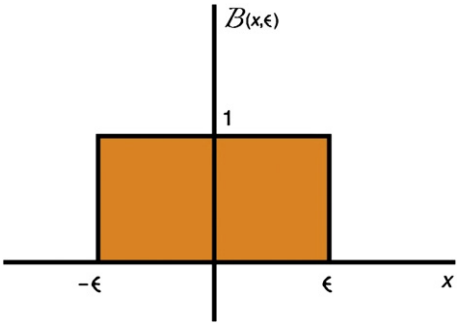

Fig. 1. The boxcar function $\mathcal{B}(x,\epsilon)$ is equal to one for $-\epsilon \leqslant x \leqslant \epsilon$ and zero otherwise.

$$tolerance = \frac{configuration_{n+1} - configuration_n}{2} \tag{8}$$

and so, if each $configuration_n$ is given by a non-negative integer number, $tolerance = 1/2$. If each configuration is instead given by a rational number separated a distance $1/d$ from the next, then $tolerance = 1/(2d)$. In Section 4.3 an example is given where the tolerance has this latter form.

The structure of Eq. (7) matches the one of simple functions found in measure theory (i.e. a function that can take a countable number of values) [25]. The independent construction presented here leads to a different interpretation of such an abstract mathematical structure within the mathematical formalism of $\mathcal{B}$-calculus, and the latter is useful to describe rule-based dynamical systems. This is the core idea of this Letter. The non-standard terms *configuration* and *tolerance* are used here because of their appealing visual character and physical flavor and to emphasize the finiteness of the sets which cover the (generally continuous) range of values that the argument x can take. Eq. (7) can be further generalized within $\mathcal{B}$-calculus to allow for non-exclusive configurations (multivalued functions), vectorial and tensorial outputs, etc.

3. General theory for cellular automata (CAs)

3.1. Universal map for one-dimensional CAs

It can be now shown how CA rules always fit in the above structure, provided by Eq. (7) and some of its possible generalizations. A spatially extended system consisting of a lattice of discrete sites is now considered. The topology and dimensions of this lattice can be arbitrary. The following discussion focus first in the case of a 1D ring containing a total number of N_s sites and it is later generalized to higher-dimensional systems.

An input is given as initial condition in the form of a vector $\mathbf{x}_0 = (x_0^1, \ldots, x_0^{N_s})$. Each of the x_0^i is an integer in the range 0 through $p-1$ (where p specifies the total number of symbols in the alphabet). The subindex labels the discrete time step in the evolution and the superindex specifies the position of the site on a 1D ring. At each time step t the vector $\mathbf{x}_t = (x_t^1, \ldots, x_t^{N_s})$ specifies the state of the CA. All inputs and outputs from the rule (i.e. all x's regardless of the labels) are always integers on the range 0 through $p-1$. Periodic boundary conditions are assumed so that $x_t^{N_s+1} = x_t^1$ and $x_t^0 = x_t^{N_s}$. Let x_{t+1}^i be taken to denote the value of site i at time step $t+1$. Formally, its dependence on the values at the previous time step is given through the mapping

$$x_{t+1}^i = \phi\left(x_t^{i-r}, x_t^{i-r+1}, \ldots, x_t^i, \ldots, x_t^{i+l-1}, x_t^{i+l}\right) \tag{9}$$

where $\phi(\cdots)$ is the function of the site values which specifies the local rule. Here r and l denote the number of cells to the right and to the left of site i respectively. The range ρ of the rule is defined as the total number of sites involved in the rule and is therefore given by $\rho = l + r + 1$. Since each site i can have p different values there is a total of $p^\rho = p^{l+r+1}$ different configurations on the table for each possible combination of symbols. For a range ρ there exists a total number of $\Gamma = p^{p^{l+r+1}}$ different rules that can be defined in this way. Each $configuration_n$ in the table is indexed by the integer number n which runs between 0 and $p^{l+r+1}-1$ (and hence $tolerance = 1/2$) and which is given as a function of the site values in the table x^i as $n = \sum_{k=-r}^{l} p^{k+r} x^{i+k}$. The x^i specify all possible configurations. The above n specifies each *static* configuration given on the table. They compare to the dynamical configuration n_t reached by site i and its r and l first neighbors at time t, $n_t = \sum_{k=-r}^{l} p^{k+r} x_t^{i+k}$. The latter is the *input* of the rule.

The outputs a_n for each configuration n are integers between 0 and $p-1$. An integer number R can then be given to specify the code of the rule $R = \sum_{n=0}^{p^{l+r+1}-1} a_n p^n$, i.e. the a_n are the coefficients that accompany the powers of p when writing R in base p. Since the rule depends not only on the a_n but also on the number of symbols p involved and on the number of neighbors to the left l and to the right r, to define an arbitrary CA rule in an unambiguous way the quantity ${}^lR_p^r$ can be introduced. It denotes symbolically in all the following the same as ϕ in Eq. (9) above. The labels l, r and p accompanying R specify the rule completely and Eq. (9) can be written in a compact and unambiguous way as

$$x_{t+1}^i = {}^lR_p^r\left(x_t^i\right) \tag{10}$$

with the understanding that the rule depends not only on x_t^i but also on the first r neighbors to the right and the first l neighbors to the left and that each of the sites can have p different values. For example, Wolfram's rule 110, involves $p=2$ symbols, and one left and one right neighbors, i.e. $r=l=1$ (see Fig. 2). Such rule is then written as ${}^1110_2^1$.

The following set of correspondences between the quantities here defined and the ones introduced in Eq. (7) in Section 2 hold

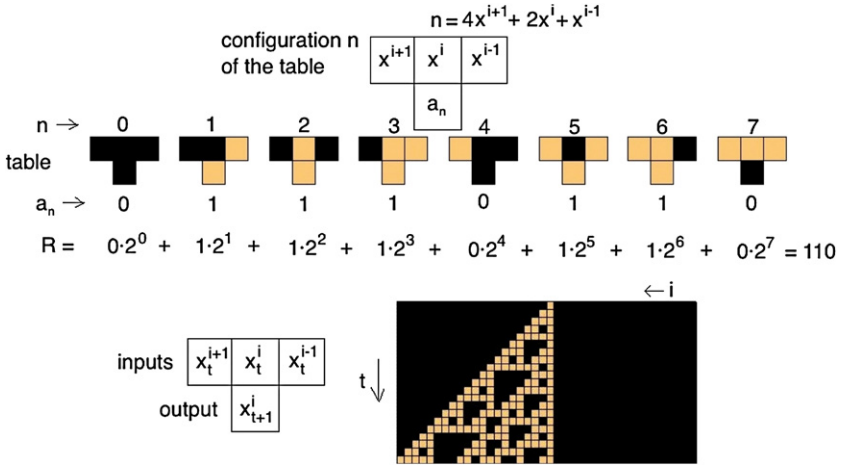

Fig. 2. Sketch of Wolfram's rule ${}^1110^1_2$. The table of the rule with each *configuration*$_n$ (n) and each *output*$_n$ (a_n) and code R is shown, as well as the spatiotemporal evolution of the CA starting from an initial condition consisting on a single site with value '1' and all others with value '0'.

$$\begin{aligned}
&\textit{input} \iff n_t = \sum_{k=-r}^{l} p^{k+r} x_t^{i+k}\\
&\textit{configuration}_n \iff n\left(\equiv \sum_{k=-r}^{l} p^{k+r} x^{i+k}\right)\\
&\textit{tolerance} \iff 1/2\\
&\textit{output}_n \iff a_n\\
&\textit{output} \iff x_{t+1}^i
\end{aligned}$$

and the universal map governing the dynamics of 1D cellular automata, by using Eq. (7), has then the form

$$x_{t+1}^i = \sum_{n=0}^{p^{l+r+1}-1} a_n \mathcal{B}\left(n - \sum_{k=-r}^{l} p^{k+r} x_t^{i+k}, \frac{1}{2}\right) \tag{11}$$

or, equivalently, by using Definition 1

$$x_{t+1}^i = \sum_{n=0}^{p^{l+r+1}-1} \frac{a_n}{2}\left(\frac{\frac{1}{2}+n-\sum_{k=-r}^{l} p^{k+r} x_t^{i+k}}{\left|\frac{1}{2}+n-\sum_{k=-r}^{l} p^{k+r} x_t^{i+k}\right|} + \frac{\frac{1}{2}-n+\sum_{k=-r}^{l} p^{k+r} x_t^{i+k}}{\left|\frac{1}{2}-n+\sum_{k=-r}^{l} p^{k+r} x_t^{i+k}\right|}\right) \tag{12}$$

Although Eq. (11) seems rather complex, it is to be noted that it is the normal form for *all first-order-in-time deterministic CA maps in 1D*. This expression makes explicit how CA behavior depends on the specification of the rule (given by the vector containing the coefficients a_n), the dimensions of the neighborhood (parameters l and r) and the total number of symbols in the alphabet p. Previous formulations [5–7] were equivalent to Eq. (9), which is clearly unsatisfactory for further mathematical analysis, since such equation is merely useful to define a CA in a formal way, but it does not yet expose the dependence of CA behavior on the different parameters. Eq. (11) also takes advantage over any kind of descriptive statement of the CA update rule, since it provides a systematic means to obtain the orbit of the CA, i.e. its trajectory in phase space, in the cases where it can be given in mathematically closed form. This is shown in Section 4.

3.2. Totalistic rules

Sometimes, the configuration is specified not in terms of its binary code but over a sum carried over all sites in the neighborhood. The output is made in such cases dependent only on the sum of the previous values. These CAs are called *totalistic* and they are a subset of the total possibilities described by Eq. (11). Each *configuration*$_n$ in the table is labeled in such a case with an integer number s which runs between 0 and $\rho(p-1)$ (the latter value is the maximum value that the sum over the site values can attain) $s = \sum_{k=-r}^{l} x^{i+k}$. The x^i have as before values between 0 and $p-1$ and specify all possible configurations of the symbols. The above s specifies each *static* configuration given on the table. They compare to the dynamical configuration s_t reached by site i and its r and l first neighbors at time t, $s_t = \sum_{k=-r}^{l} x_t^{i+k}$ which is the *input* of the totalistic rule.

The outputs σ_s for each configuration s have integer values between 0 and $p-1$ like the inputs and the output of the rule. An integer number R can then be given to specify the code of the totalistic rule in the following way $R = \sum_{s=0}^{\rho(p-1)} \sigma_s p^s$. A totalistic CA rule can be therefore labeled as ${}^lRT^r_p$, where the label T is added to avoid confusion with the coding in the normal form. The following set of correspondences between these quantities and the ones introduced in Eq. (7) in Section 2 holds now

$$\textit{input} \iff s_t = \sum_{k=-r}^{l} x_t^{i+k}$$

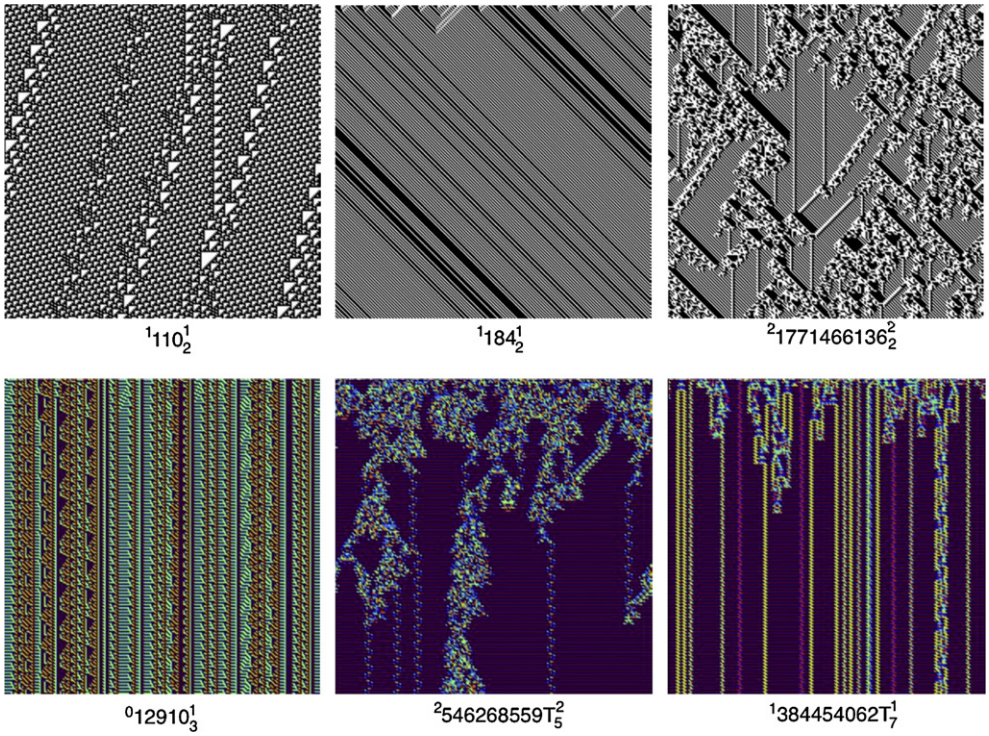

Fig. 3. Spatiotemporal evolution of rules ${}^{l}R^{r}_{p}$ and ${}^{l}RT^{r}_{p}$ indicated in the figure obtained from Eqs. (11) and (13) respectively for a system size $N_s = 200$ and 200 time steps (time flows from top to bottom) for a random initial condition.

$$\begin{aligned} \textit{configuration}_n &\iff s\left(\equiv \sum_{k=-r}^{l} x^{i+k}\right) \\ \textit{tolerance} &\iff 1/2 \\ \textit{output}_n &\iff \sigma_s \\ \textit{output} &\iff x^{i}_{t+1} \end{aligned}$$

And therefore, the universal map governing the dynamics of 1D totalistic cellular automata, is obtained by using Eq. (7) as

$$x^{i}_{t+1} = \sum_{s=0}^{\rho(p-1)} \sigma_s \mathcal{B}\left(s - \sum_{k=-r}^{l} x^{i+k}_{t}, \frac{1}{2}\right) \tag{13}$$

Eq. (13) is a particular case of Eq. (11). Starting from a totalistic rule with vector $(\sigma_0, \sigma_1, \ldots, \sigma_{\rho(p-1)})$ described by Eq. (13) the vector specifying the normal rule as described by Eq. (11) $(a_0, a_1, \ldots, a_{p^{l+r+1}-1})$ can be calculated from the following system of equations (for all possible integer values of $k \in [-r, l]$ and $x^{i+k} \in [0, p-1]$)

$$n = \sum_{k=-r}^{l} p^{k+r} x^{i+k} \qquad s = \sum_{k=-r}^{l} x^{i+k} \qquad a_n = \sigma_s \tag{14}$$

Eq. (11) or its simplified version for totalistic rules Eq. (13) can be used to follow the spatiotemporal evolution of *all* deterministic 1D cellular automata which are first order in time. Fig. 3 shows some examples, where Eqs. (11) and (13) are run on a ring with $N_s = 200$ sites, after giving values for r, l, p and R. As discussed above, the latter quantity specifies the vector $(a_0, a_1, \ldots, a_{p^{l+r+1}-1})$, since it provides the decimal representation of the base p number $a_{p^{l+r+1}-1} \cdots a_1 a_0$. (In the case of totalistic rules, the number R provides the vector $(\sigma_0, \sigma_1, \ldots, \sigma_{\rho(p-1)})$, since it coincides with the decimal representation of the base p number $\sigma_{\rho(p-1)} \cdots \sigma_1 \sigma_0$.)

By introducing appropriate products of boxcar functions in the r.h.s. of Eqs. (11) and (13), these maps can be straightforwardly generalized to any order in time and an arbitrary number of dimensions. In Sections 3.4 and 3.5 these generalizations are discussed.

3.3. Universal cellular automata maps in polynomial form

Before the first iteration takes place, Eqs. (11) and (13) also accept *real* numbers as *input*. This is a very desirable property, if one thinks in practical applications where Eqs. (11) and (13) may be implemented in, for example, experiments involving electric circuits. These real inputs are rounded by the boxcar function to the closest integer in the first iteration and the first *output* is therefore an integer number. The subsequent iterations of these maps take place entirely on the integers between 0 and $p-1$.

If the initial *input* values are already given as integers between 0 and $p-1$, Eqs. (11) and (13) can be written in polynomial form: since the dynamics remain always from the very beginning on these integers, the polynomial interpolation of the r.h.s. of Eqs. (11) and

(13) provides an equivalent dynamical description. The following theorem allows to construct such a polynomial. This is an important result since it allows to pass from a description in terms of boxcar functions to polynomials involving integer variables. In order to obtain the latter a linear system of equations involving a Vandermonde matrix of $\Omega \times \Omega$ dimensions (where $\Omega = p^{l+r+1}$) needs to be solved. Eq. (11), which is computationally inexpensive and sufficient to characterize every CA rule is, therefore, more fundamental.

Theorem *(Polynomial maps). The following result holds*

$$ {}^{l}R_{p}^{r}\left(x_t^i\right) \equiv x_{t+1}^i = \sum_{n=0}^{p^{l+r+1}-1} \alpha_n \left(\sum_{k=-r}^{l} p^{k+r} x_t^{i+k} \right)^n \tag{15} $$

where the coefficients α_n's (rational numbers) are the solutions of the following linear system of equations

$$ \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ 2^{\Omega-1} & 2^{\Omega-2} & 2^{\Omega-3} & \dots & 2 & 1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ (\Omega-1)^{\Omega-1} & (\Omega-1)^{\Omega-2} & (\Omega-1)^{\Omega-3} & \dots & \Omega-1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_{\Omega-1} \\ \alpha_{\Omega-2} \\ \alpha_{\Omega-3} \\ \vdots \\ \alpha_0 \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{\Omega-1} \end{bmatrix} \tag{16} $$

where $\Omega = p^{l+r+1}$. For a totalistic rule, one has

$$ {}^{l}RT_{p}^{r}\left(x_t^i\right) \equiv x_{t+1}^i = \sum_{s=0}^{\rho(p-1)} \alpha_s \left(\sum_{k=-r}^{l} x_t^{i+k} \right)^s \tag{17} $$

where the coefficients α_s's (rational numbers) are the solutions of the following linear system of equations

$$ \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ 2^{\Theta} & 2^{\Theta-1} & 2^{\Theta-2} & \dots & 2 & 1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \Theta^{\Theta} & \Theta^{\Theta-1} & \Theta^{\Theta-2} & \dots & \Theta & 1 \end{bmatrix} \begin{bmatrix} \alpha_{\Theta} \\ \alpha_{\Theta-1} \\ \alpha_{\Theta-2} \\ \vdots \\ \alpha_0 \end{bmatrix} = \begin{bmatrix} \sigma_0 \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_{\Theta} \end{bmatrix} \tag{18} $$

where $\Theta = \rho(p-1)$.

Proof. Eqs. (15) and (17) are, respectively, the polynomial interpolations of Eqs. (11) and (13). For the topic of polynomial interpolation see for example [26]. □

When x is a boolean variable (i.e. a variable that can take values either zero or one), one has $x^2 = x$, $(1-x)^2 = 1-x$ and $x(1-x) = 0$. By using these simple relationships and the above theorem, it is easy to prove the following corollaries. (All three corollaries can also be directly proved from Eq. (11) without need of using the theorem above. Such alternative proof is given in detail in [27].)

Corollary 1 *(Universal CA map for the 4 most simple boolean rules). The universal CA map, Eq. (11), reduces to the following expression*

$$ x_{t+1}^i = a_0\left(1 - x_t^i\right) + a_1 x_t^i \tag{19} $$

for the 4 most simple boolean rules ($p = 2, l = 0, r = 0$), i.e. the rules ${}^{0}R_{2}^{0}$, where $R = \sum_{n=0}^{1} a_n 2^n$.

Corollary 2 *(Universal CA map for the 16 boolean logical functions). The universal CA map, Eq. (11), reduces to the following expression (boolean logical functions)*

$$ x_{t+1}^i = a_0\left(1 - x_t^i\right)\left(1 - x_t^{i-1}\right) + a_1 x_t^{i-1}\left(1 - x_t^i\right) + a_2 x_t^i\left(1 - x_t^{i-1}\right) + a_3 x_t^i x_t^{i-1} \tag{20} $$

for all 16 rules with $p = 2, l = 0, r = 1$, i.e. the rules ${}^{0}R_{2}^{1}$, and to

$$ x_{t+1}^i = a_0\left(1 - x_t^{i+1}\right)\left(1 - x_t^i\right) + a_1 x_t^i\left(1 - x_t^{i+1}\right) + a_2 x_t^{i+1}\left(1 - x_t^i\right) + a_3 x_t^{i+1} x_t^i \tag{21} $$

for all 16 rules with $p = 2, l = 1, r = 0$, i.e. the rules ${}^{1}R_{2}^{0}$. In both cases $R = \sum_{n=0}^{3} a_n 2^n$.

Corollary 3 *(Universal CA map for the 256 boolean Wolfram rules). The universal CA map, Eq. (11), reduces to the following expression*

$$ \begin{aligned} x_{t+1}^i = {} & a_0\left(1 - x_t^{i+1}\right)\left(1 - x_t^i\right)\left(1 - x_t^{i-1}\right) + a_1 x_t^{i-1}\left(1 - x_t^{i+1}\right)\left(1 - x_t^i\right) \\ & + a_2 x_t^i\left(1 - x_t^{i+1}\right)\left(1 - x_t^{i-1}\right) + a_3 x_t^i x_t^{i-1}\left(1 - x_t^{i+1}\right) + a_4 x_t^{i+1}\left(1 - x_t^i\right)\left(1 - x_t^{i-1}\right) \\ & + a_5 x_t^{i+1} x_t^{i-1}\left(1 - x_t^i\right) + a_6 x_t^{i+1} x_t^i\left(1 - x_t^{i-1}\right) + a_7 x_t^{i+1} x_t^i x_t^{i-1} \end{aligned} \tag{22} $$

for all 256 rules with $p = 2, l = 1, r = 1$ (Wolfram rules), i.e. the rules ${}^{1}R_{2}^{1}$, where $R = \sum_{n=0}^{7} a_n 2^n$.

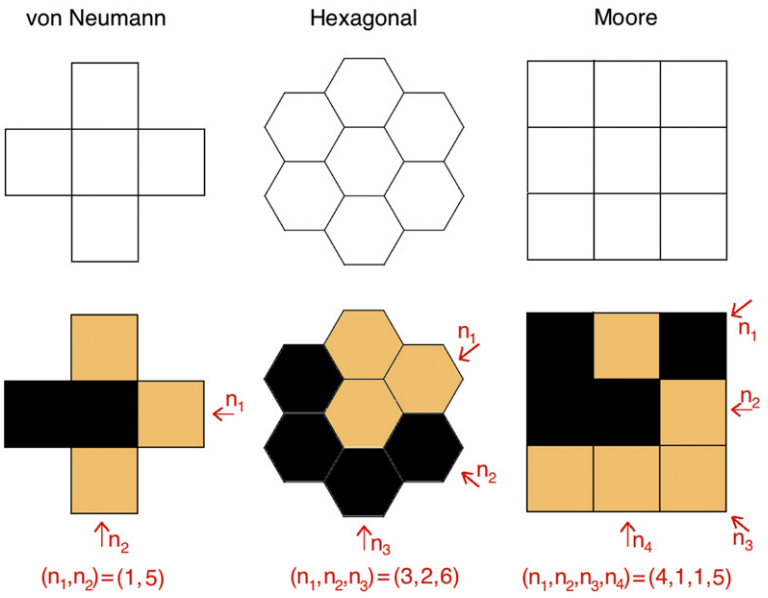

Fig. 4. Symmetrical von Neumann, hexagonal and Moore neighborhoods in two dimensions. A configuration can be specified by giving two, three, or four 1D codes, respectively.

Examples. Rule ${}^{0}6^{1}_{2}$ has vector $(a_0, a_1, a_2, a_3) = (0, 1, 1, 0)$ and therefore its polynomial map is given by

$$x^i_{t+1} = x^{i-1}_t + x^i_t - 2x^i_t x^{i-1}_t \tag{23}$$

Clearly, the output in the l.h.s. provides the XOR logical function of the two site values at the previous time. The same logical function is implemented by rule ${}^{1}6^{0}_{2}$. Rule ${}^{0}14^{1}_{2}$ has vector $(a_0, a_1, a_2, a_3) = (0, 1, 1, 1)$ and therefore its polynomial map is given by

$$x^i_{t+1} = x^{i-1}_t + x^i_t - x^i_t x^{i-1}_t \tag{24}$$

and now the rule implements the OR logical function of the two site values. Since Wolfram's famous rule ${}^{1}110^{1}_{2}$ has vector $(a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7) = (0, 1, 1, 1, 0, 1, 1, 0)$ (see Fig. 2), from Eq. (22) its map has the remarkably simple expression

$$x^i_{t+1} = x^i_t + x^{i-1}_t\left(1 - x^i_t - x^i_t x^{i+1}_t\right) \tag{25}$$

as can be cursorily checked by replacing all possible site values in the r.h.s. Remarkably, the site x^{i+1}_t tunes the logical function implemented in the other two sites between *OR* and *XOR*, when the site value is '0' or '1', respectively.

3.4. CAs in higher dimensions

The above results can be easily generalized to an arbitrary number of dimensions. In 2D, for example, by using the site that is updated after each time step as a pivoting site and defining 1D codes for each spatial direction counterclockwise, universal maps for all possible deterministic CAs in 2D (depending on the topology of the interactions in the lattice) can be derived. The most popular neighborhoods in 2D are shown in Fig. 4. The von Neumann neighborhood can be specified giving two binary codes n_1 and n_2 as indicated in the figure. The hexagonal neighborhood requires the specification of three codes to describe each configuration. Finally, the Moore neighborhood requires four 1D codes. In each neighborhood, the site on the center is updated in the next time step.

Since a configuration is now composed of several 1D configurations that must hold simultaneously, a product of boxcar functions is needed to fully account for the dynamical state. For a von Neumann neighborhood with horizontal and vertical ranges given by the pairs $\rho_1 = l_1 + r_1 + 1$ and $\rho_2 = l_2 + r_2 + 1$ respectively, it is obtained, instead of Eq. (11)

$$x^{i,j}_{t+1} = \sum_{n_1=0}^{p^{l_1+r_1+1}-1} \sum_{n_2=0}^{p^{l_2+r_2+1}-1} a_{n_1,n_2} \mathcal{B}\left(n_1 - \sum_{k=-r_1}^{l_1} p^{k+r_1} x^{i+k,j}_t, \frac{1}{2}\right) \mathcal{B}\left(n_2 - \sum_{k=-r_2}^{l_2} p^{k+r_2} x^{i,j+k}_t, \frac{1}{2}\right) \tag{26}$$

(note that in the figure only the entirely symmetrical neighborhood with $\rho_1 = \rho_2 = 3$ is shown). For a totalistic rule over a von Neumann neighborhood

$$x^{i,j}_{t+1} = \sum_{s=0}^{(\rho_1+\rho_2-1)(p-1)} \sigma_s \mathcal{B}\left(s - \sum_{k=-r_1}^{l_1} x^{i+k,j}_t - \sum_{k=-r_2}^{l_2} x^{i,j+k}_t + x^{i,j}_t, \frac{1}{2}\right) \tag{27}$$

For an hexagonal neighborhood

$$x_{t+1}^{i,j} = \sum_{n_1=0}^{p^{l_1+r_1+1}-1} \sum_{n_2=0}^{p^{l_2+r_2+1}-1} \sum_{n_3=0}^{p^{l_3+r_3+1}-1} a_{n_1,n_2,n_3} \mathcal{B}\left(n_1 - \sum_{k=-r_1}^{l_1} p^{k+r_1} x_t^{i+k,j-k}, \frac{1}{2}\right) \times \mathcal{B}\left(n_2 - \sum_{k=-r_2}^{l_2} p^{k+r_2} x_t^{i+k,j+k}, \frac{1}{2}\right) \mathcal{B}\left(n_3 - \sum_{k=-r_3}^{l_3} p^{k+r_3} x_t^{i,j+k}, \frac{1}{2}\right) \tag{28}$$

and for the totalistic case

$$x_{t+1}^{i,j} = \sum_{s=0}^{(\rho_1+\rho_2+\rho_3-2)(p-1)} \sigma_s \mathcal{B}\left(s + 2x_t^{i,j} - \sum_{k=-r_1}^{l_1} x_t^{i+k,j-k} - \sum_{k=-r_2}^{l_2} x_t^{i+k,j+k} - \sum_{k=-r_3}^{l_3} x_t^{i,j+k}, \frac{1}{2}\right) \tag{29}$$

Finally, for the Moore neighborhood one has, for the general case,

$$x_{t+1}^{i,j} = \sum_{n_1=0}^{p^{l_1+r_1+1}-1} \sum_{n_2=0}^{p^{l_2+r_2+1}-1} \sum_{n_3=0}^{p^{l_3+r_3+1}-1} \sum_{n_4=0}^{p^{l_4+r_4+1}-1} a_{n_1,n_2,n_3,n_4} \mathcal{B}\left(n_1 - \sum_{k=-r_1}^{l_1} p^{k+r_1} x_t^{i+k,j-k}, \frac{1}{2}\right) \times \mathcal{B}\left(n_2 - \sum_{k=-r_2}^{l_2} p^{k+r_2} x_t^{i+k,j}, \frac{1}{2}\right) \mathcal{B}\left(n_3 - \sum_{k=-r_3}^{l_3} p^{k+r_3} x_t^{i+k,j+k}, \frac{1}{2}\right) \times \mathcal{B}\left(n_4 - \sum_{k=-r_4}^{l_4} p^{k+r_4} x_t^{i,j+k}, \frac{1}{2}\right) \tag{30}$$

and for the totalistic one

$$x_{t+1}^{i,j} = \sum_{s=0}^{(\rho_1+\rho_2+\rho_3+\rho_4-3)(p-1)} \sigma_s \mathcal{B}\left(s + 3x_t^{i,j} - \sum_{k=-r_1}^{l_1} x_t^{i+k,j-k} - \sum_{k=-r_2}^{l_2} x_t^{i+k,j} - \sum_{k=-r_3}^{l_3} x_t^{i+k,j+k} - \sum_{k=-r_4}^{l_4} x_t^{i,j+k}, \frac{1}{2}\right) \tag{31}$$

Above the most general situation, where each spatial direction can be non-symmetrical respect to the site i, j, was considered. In most interesting cases, however, symmetrical neighborhoods as the ones in Fig. 4 are considered, when particularized to them, these expressions reduce to very simple forms. Taking now, for example, a Moore neighborhood with $l_1 = l_2 = l_3 = l_4 = r_1 = r_2 = r_3 = r_4 = 1$, as in Fig. 4, the latter expression simplifies to

$$x_{t+1}^{i,j} = \sum_{s=0}^{9} \sigma_s \mathcal{B}\left(s - \sum_{k,m=-1}^{1} x_t^{i+k,j+m}, \frac{1}{2}\right) \tag{32}$$

A famous totalistic CA in such a Moore neighborhood due to Gerard Vichniac is the so-called "vote" and is given by a vector $(\sigma_0, \sigma_1, \ldots, \sigma_9) = (0,0,0,0,1,1,1,1,1,1)$. For such CA, since the sum over the cells in the neighborhood can only be at most '9' and the CA only returns '1' when $s \geqslant 5$ and zero otherwise,

$$x_{t+1}^{i,j} = \sum_{s=5}^{9} \mathcal{B}\left(s - \sum_{k,m=-1}^{1} x_t^{i+k,j+m}, \frac{1}{2}\right) = \sum_{s=5}^{\infty} \mathcal{B}\left(s - \sum_{k,m=-1}^{1} x_t^{i+k,j+m}, \frac{1}{2}\right) \tag{33}$$

where the sum has been extended to infinity (a trick that is possible to employ here for this specific CA because of its structure). The sum can be now carried explicitly to give

$$x_{t+1}^{i,j} = H\left(\sum_{k,m=-1}^{1} x_t^{i+k,j+m} - \frac{9}{2}\right) \tag{34}$$

where $H(x)$ is the Heaviside function. The r.h.s. of the latter expression is only one when the sum over the cells is higher or equal than 5.

Another famous semitotalistic 2D boolean CA called "Game of life" [28,29] and invented by J.H. Conway is also defined on such a Moore neighborhood in terms of the following rules governing the behavior of the site in the center of the neighborhood (this site is called a "live cell" if it has value '1' and a "dead cell" if it has value '0')

1. Any live cell with fewer than two live neighbors dies, as if caused by under-population.
2. Any live cell with two or three live neighbors lives on to the next generation.
3. Any live cell with more than three live neighbors dies, as if by overcrowding.
4. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

This set of rules can indeed be reduced to three

1. If on the entire neighborhood the sum of all site values is '3' the cell in the center lives (or becomes a live cell) in the next generation.
2. If on the entire neighborhood the sum of all site values is '4' the cell in the center lives in the next generation only if it is already alive.

3. The cell in the center is dead on the next generation if neither 1. nor 2. holds

By employing Eq. (7) a map for this CA can be written *directly*, translating these rules into the mathematical language of $\mathcal{B}$-calculus

$$x_{t+1}^{i,j} = \mathcal{B}\left(3 - \sum_{k,m=-1}^{1} x_t^{i+k,j+m}, \frac{1}{2}\right) + x_t^{i,j}\mathcal{B}\left(4 - \sum_{k,m=-1}^{1} x_t^{i+k,j+m}, \frac{1}{2}\right) \tag{35}$$

This is a local map for each site i, j in the lattice that implements the Game of Life.

For 3D lattices, the expressions above can easily be further generalized. For example, for a totalistic CA in a symmetric cubic 3D lattice, the universal map takes the simple form

$$x_{t+1}^{i,j,h} = \sum_{s=0}^{(p-1)R^3} \sigma_s \mathcal{B}\left(s - \sum_{k,m,q=-(R-1)/2}^{(R-1)/2} x_t^{i+k,j+m,h+q}, \frac{1}{2}\right) \tag{36}$$

where R is the number of boxes in one of the edges of the cube (an odd number).

3.5. CAs of higher order in time and time-reversible CA

If the output of a 1D CA depends on T previous configurations of the i site, a "spatial" and "temporal" codes can be given and the resulting expression for such a 1D CA of higher order in time can then be written as

$$x_{t+1}^{i} = \sum_{n=0}^{p^{l+r+1}-1} \sum_{m=0}^{p^{T}-1} a_{n,m} \mathcal{B}\left(n - \sum_{k=-r}^{l} p^{k+r} x_t^{i+k}, \frac{1}{2}\right) \mathcal{B}\left(m - \sum_{k=1}^{T} p^{k-1} x_{t-k}^{i}, \frac{1}{2}\right) \tag{37}$$

For each CA rule ${}^{l}R_p^r(x_t^i)$ described by Eq. (11) a reversible rule, which can be denoted as $\mathrm{rev}[{}^{l}R_p^r](x_t^i)$ can be constructed as

$$x_{t+1}^{i} = \sum_{n=0}^{p^{l+r+1}-1} \left|a_n - x_{t-1}^{i}\right| \mathcal{B}\left(n - \sum_{k=-r}^{l} p^{k+r} x_t^{i+k}, \frac{1}{2}\right) \tag{38}$$

It can be easily proven that these CA rules exhibit the same dynamical behavior to the future and to the past (the absolute value in the r.h.s. of Eq. (38) warrants the symmetry under time reversal).

4. Discussion

4.1. Inductive method for CAs

Eq. (11) and its totalistic version Eq. (13) (as well as all its generalized versions to higher dimensions and higher order in time) can be used to find the mathematical expressions for the orbits of CAs, provided they exist. Such orbit gives the value of x_t^i, for each position i as a function of t and the initial condition $(x_0^1, \ldots, x_0^i, \ldots, x_0^{N_s})$. The method to find the orbit proceeds then through the following steps

1. For a given initial condition $(x_0^1, \ldots, x_0^i, \ldots, x_0^{N_s})$ by using Eq. (11) calculate, for a certain position i, the iterates $x_1^i, x_2^i, \ldots, x_{l+r+1}^i$ at later times.
2. Derive from such iterates the general expression for the orbit $x_t^i = f(x_0^1, \ldots, x_0^i, \ldots, x_0^{N_s}, t)$.
3. Assume that the expression for the orbit is valid at an arbitrary time t.
4. Use Eq. (11) to prove that the expression for the orbit is then valid at time $t+1$.

The CA rule ${}^{0}3_2^1$ is now considered as an example to apply this method. For this rule $(a_0, a_1, a_2, a_3) = (1, 1, 0, 0)$ and, therefore, from Eq. (11)

$$x_{t+1}^{i} = \mathcal{B}\left(-x_t^{i-1} - 2x_t^{i}, \frac{1}{2}\right) + \mathcal{B}\left(1 - x_t^{i-1} - 2x_t^{i}, \frac{1}{2}\right) = 1 - x_t^{i} \tag{39}$$

where it has been considered that the x's take only integer values and Corollary 2 has been used (the validity of the second equality can be cursorily checked by replacing both possible values '0' and '1' for x_t^i and x_t^{i-1}). The inductive method for CAs can now be applied. In a position i the first iterates of the map starting from an arbitrary initial condition $(x_0^1, \ldots, x_0^i, \ldots, x_0^{N_s})$ are

$$x_1^i = 1 - x_0^i \tag{40}$$

$$x_2^i = 1 - x_1^i = 1 - \left(1 - x_0^i\right) = x_0^i \tag{41}$$

This completes Step 1 of the inductive method. From these two iterates it can be seen that at time t odd, the i site has value $1 - x_0^i$ but when t is even, the i site has value x_0^i. A general expression that summarizes these results for arbitrary time is thus

$$x_t^i = (-1)^t x_0^i + \frac{1}{2}\left(1 - (-1)^t\right) \tag{42}$$

This completes Step 2. This expression is now assumed to be valid at an arbitrary time t (Step 3). At time $t+1$ then, by using the map for this rule, Eq. (39),

$$x_{t+1}^i = 1 - x_t^i = 1 - \left[(-1)^t x_0^i + \frac{1}{2}\left(1-(-1)^t\right)\right] = -(-1)^t x_0^i + \frac{1}{2}\left(1+(-1)^t\right)$$
$$= (-1)^{t+1} x_0^i + \frac{1}{2}\left(1-(-1)^{t+1}\right) \tag{43}$$

which proves by induction that Eq. (42) is a valid mathematical expression for the orbit for this specific CA rule (Step 4).

By using the above method one can prove that all rules ${}^0R_2^1$ and ${}^1R_2^0$ (and hence all rules ${}^0R_2^0$ as well) can be solved for the orbit. The following theorem is given without proof since all cases can be obtained by using the inductive method.

Theorem *(Predictability of all boolean rules ${}^0R_2^1$ and ${}^1R_2^0$). If $\mathbf{x}_0 = (x_0^1, \ldots, x_0^i, \ldots, x_0^{N_s})$ denotes the initial state of a CA at time $t=0$, for time $t \geqslant 1$ (positive integer), all boolean rules ${}^0R_2^1$ and ${}^1R_2^0$ are predictable and the value at the site on a later time t, x_t^i, can be known for each rule, as a function of t and the initial site values $\mathbf{x}_0$. The orbits (local dependence in time for the state of the cellular automata) for all rules ${}^0R_2^1$ and ${}^1R_2^0$ are*

- ${}^0 0_2^1$: $x_t^i = 0$
- ${}^0 1_2^1$: $x_t^i = \frac{1}{2}\left(1-(-1)^t\right)\left(1 - x_0^{i-\frac{t-1}{2}} - x_0^{i-\frac{t+1}{2}} + x_0^{i-\frac{t-1}{2}} x_0^{i-\frac{t+1}{2}}\right) + \frac{1}{2}\left(1+(-1)^t\right)\left(x_0^{i-\frac{t}{2}} + x_0^{i-\frac{t-2}{2}} x_0^{i-\frac{t+2}{2}} - x_0^{i-\frac{t}{2}} x_0^{i-\frac{t-2}{2}} x_0^{i-\frac{t+2}{2}}\right)$
- ${}^0 2_2^1$: $x_t^i = x_0^{i-t} - x_0^{i-t} x_0^{i-t+1}$
- ${}^0 3_2^1$: $x_t^i = (-1)^t x_0^i + \frac{1}{2}\left(1-(-1)^t\right)$
- ${}^0 4_2^1$: $x_t^i = x_0^i - x_0^i x_0^{i-1}$
- ${}^0 5_2^1$: $x_t^i = (-1)^t x_0^{i-t} + \frac{1}{2}\left(1-(-1)^t\right)$
- ${}^0 6_2^1$: $x_t^i = \sum_{m=0}^{2^{t-1}} \mathcal{B}\left(2m+1-\sum_{k=0}^{t} \frac{t!}{k!(t-k)!} x_0^{i-t+k}, \frac{1}{2}\right)$
- ${}^0 7_2^1$: $x_t^i = \frac{1}{2}\left(1-(-1)^t\right)\left(1 - x_0^{i-\frac{t-1}{2}} x_0^{i-\frac{t+1}{2}}\right) + \frac{1}{2}\left(1+(-1)^t\right)\left(x_0^{i-\frac{t}{2}} x_0^{i-\frac{t-2}{2}} + x_0^{i-\frac{t}{2}} x_0^{i-\frac{t+2}{2}} - x_0^{i-\frac{t}{2}} x_0^{i-\frac{t-2}{2}} x_0^{i-\frac{t+2}{2}}\right)$
- ${}^0 8_2^1$: $x_t^i = \prod_{k=0}^{t} x_0^{i-k}$
- ${}^0 9_2^1$: $x_t^i = \sum_{m=0}^{2^{t-1}} \mathcal{B}\left(2m-\sum_{k=0}^{t} \frac{t!}{k!(t-k)!}\left(1-x_0^{i-t+k}\right), \frac{1}{2}\right)$
- ${}^0 10_2^1$: $x_t^i = x_0^{i-t}$
- ${}^0 11_2^1$: $x_t^i = 1 - x_0^{i-t+1} + x_0^{i-t} x_0^{i-t+1}$
- ${}^0 12_2^1$: $x_t^i = x_0^i$
- ${}^0 13_2^1$: $x_t^i = 1 - x_0^{i-1} + x_0^i x_0^{i-1}$
- ${}^0 14_2^1$: $x_t^i = 1 - \prod_{k=0}^{t}\left(1 - x_0^{i-k}\right)$
- ${}^0 15_2^1$: $x_t^i = 1$
- ${}^1 0_2^0$: $x_t^i = 0$
- ${}^1 1_2^0$: $x_t^i = \frac{1}{2}\left(1-(-1)^t\right)\left(1 - x_0^{i+\frac{t-1}{2}} - x_0^{i+\frac{t+1}{2}} + x_0^{i+\frac{t-1}{2}} x_0^{i+\frac{t+1}{2}}\right) + \frac{1}{2}\left(1+(-1)^t\right)\left(x_0^{i+\frac{t}{2}} + x_0^{i+\frac{t-2}{2}} x_0^{i+\frac{t+2}{2}} - x_0^{i+\frac{t}{2}} x_0^{i+\frac{t-2}{2}} x_0^{i+\frac{t+2}{2}}\right)$
- ${}^1 2_2^0$: $x_t^i = x_0^i - x_0^i x_0^{i+1}$
- ${}^1 3_2^0$: $x_t^i = (-1)^t x_0^{i+t} + \frac{1}{2}\left(1-(-1)^t\right)$
- ${}^1 4_2^0$: $x_t^i = x_0^{i+t} - x_0^{i+t} x_0^{i+t-1}$
- ${}^1 5_2^0$: $x_t^i = (-1)^t x_0^i + \frac{1}{2}\left(1-(-1)^t\right)$
- ${}^1 6_2^0$: $x_t^i = \sum_{m=0}^{2^{t-1}} \mathcal{B}\left(2m+1-\sum_{k=0}^{t} \frac{t!}{k!(t-k)!} x_0^{i+t-k}, \frac{1}{2}\right)$

- ${}^{1}7_2^0$: $x_t^i = \frac{1}{2}\left(1-(-1)^t\right)\left(1-x_0^{i+\frac{t-1}{2}}x_0^{i+\frac{t+1}{2}}\right)+\frac{1}{2}\left(1+(-1)^t\right)\left(x_0^{i+\frac{t}{2}}x_0^{i+\frac{t-2}{2}}+x_0^{i+\frac{t}{2}}x_0^{i+\frac{t+2}{2}}-x_0^{i+\frac{t}{2}}x_0^{i+\frac{t-2}{2}}x_0^{i+\frac{t+2}{2}}\right)$
- ${}^{1}8_2^0$: $x_t^i = \prod_{k=0}^{t} x_0^{i+k}$
- ${}^{1}9_2^0$: $x_t^i = \sum_{m=0}^{2^{t-1}} \mathcal{B}\left(2m-\sum_{k=0}^{t}\frac{t!}{k!(t-k)!}(1-x_0^{i+t-k}),\frac{1}{2}\right)$
- ${}^{1}10_2^0$: $x_t^i = x_0^i$
- ${}^{1}11_2^0$: $x_t^i = 1-x_0^{i+1}+x_0^{i+1}x_0^i$
- ${}^{1}12_2^0$: $x_t^i = x_0^{i+t}$
- ${}^{1}13_2^0$: $x_t^i = 1-x_0^{i+t-1}+x_0^{i+t}x_0^{i+t-1}$
- ${}^{1}14_2^0$: $x_t^i = 1-\prod_{k=0}^{t}(1-x_0^{i+k})$
- ${}^{1}15_2^0$: $x_t^i = 1$.

For some of these CA rules other equivalent expressions for the orbits can be given. Since rules ${}^{0}R_2^1$ and ${}^{1}R_2^0$ represent the 16 different logical boolean operators [27], the inductive method for CAs presented here coincides with the inductive proofs of boolean expressions as found in textbooks on boolean algebra and set theory. For example, since rule ${}^{0}14_2^1$ implements the 'OR' logical function the iterates in the inductive method for this rule correspond exactly to the proof by induction of the celebrated inclusion–exclusion principle of set theory (see [30] for an exposition of this theorem). Furthermore, one can use the orbits derived for rules ${}^{0}8_2^1$ (implementing the 'AND' function) and ${}^{0}14_2^1$ to prove De Morgan' laws of boolean algebra (see [31] for a concise account of these laws). The important (potential) implication of all these exemplary correspondences is that a successful application of the induction method for CAs can be thought as isomorphic to the successful application of induction to prove any mathematical theorem, in general.

Specially interesting are the rules ${}^{0}6_2^1$, ${}^{0}9_2^1$, ${}^{1}6_2^0$, ${}^{1}9_2^0$. Although predictable, the behavior of these rules is far more complex than the one of the other rules. They correspond to the CA implementations of the XOR logical function and its complementary the XNOR function. These rules represent addition modulo 2 of both site values at a previous instant of time. Therefore, they are also totalistic rules. Rules ${}^{0}6_2^1$, ${}^{1}6_2^0$, ${}^{0}9_2^1$ and ${}^{1}9_2^0$ have totalistic codes ${}^{0}2T_2^1$, ${}^{1}2T_2^0$, ${}^{0}5T_2^1$, ${}^{1}5T_2^0$ respectively.

By using the inductive method for CAs, many Wolfram rules ${}^{1}R_2^1$ can be mathematically solved for the orbit as well, so that the solution at each site value i is known in closed form as a function of the initial condition and time. For example, rule ${}^{1}12_2^1$ has map $x_{t+1}^i = x_t^i - x_t^i x_t^{i+1}$ and the latter can be solved for the orbit, yielding $x_t^i = x_0^i - x_0^i x_0^{i+1}$ so that each site value i is known in closed form as a function of the initial condition and time. Important exceptions where the inductive method for CAs fails are discussed below.

4.2. Computational irreducibility: The failure of the inductive method for CAs

Computational irreducibility is one of the main ideas proposed by Stephen Wolfram in his book [1]. This is the term that denotes the inability to shortcut a program (e.g., a system), or to describe its behavior in a simple way. In the context of the CA theory presented in this Letter, computational irreducibility leads to the impossibility of finding a mathematical expression for the orbit of the CA.

A well-known example of computationally irreducible CA rule is Wolfram's ${}^{1}110_2^1$ rule, which has, from Corollary 2, the simple map $x_{t+1}^i = x_t^i + x_t^{i-1}(1-x_t^i - x_t^i x_t^{i+1})$. The first iterates of the CA evolution (starting from an arbitrary initial condition) are

$$x_1^i = x_0^i + x_0^{i-1}\left(1-x_0^i-x_0^i x_0^{i+1}\right) \tag{44}$$

$$x_2^i = x_0^{i-2} - x_0^{i-1}\left(x_0^{i+1}+x_0^{i-2}-1\right)+x_0^i\left[1+\left(x_0^{i+2}x_0^{i+1}-2\right)x_0^{i-2}-x_0^{i-1}\left(2-3x_0^{i-2}+x_0^{i+1}\left(-2+\left(2+x_0^{i+2}\right)x_0^{i-2}\right)\right)\right] \tag{45}$$

It can be observed that it is not possible to derive a mathematical expression for the orbit and that the second step of the inductive method cannot be completed. The reason is that for increasingly longer times, the CA behavior depends on the values of the initial condition on an increasingly larger number of sites ($2t+1$ sites) *and* it does it in a nonlinear way precluding an explicit relationship among the coefficients to be found. The only possibility of determining *exactly* the behavior for an arbitrary initial condition is to run the iterative process step by step. This is the main feature of a computationally irreducible process and the simple map derived here for Wolfram's rule ${}^{1}110_2^1$ makes evident this behavior (rendering the inductive method for CA inapplicable in this case).

The map resulting by the repeated application of a computationally irreducible rule corresponds to a computationally irreducible rule of higher range as well. For example, the map for the rule ${}^{2}1922978462_2^2$ can be obtained from Eq. (11) and has the form

$$x_{t+1}^i = x_t^{i-2} - x_t^{i-1}\left(x_t^{i+1}+x_t^{i-2}-1\right)+x_t^i\left[1+\left(x_t^{i+2}x_t^{i+1}-2\right)x_t^{i-2}-x_t^{i-1}\left(2-3x_t^{i-2}+x_t^{i+1}\left(-2+\left(2+x_t^{i+2}\right)x_t^{i-2}\right)\right)\right] \tag{46}$$

for which, clearly, one iterate corresponds to two iterates of rule ${}^{1}110_2^1$, i.e. ${}^{2}1922978462_2^2(x_t^i) = {}^{1}110_2^1[{}^{1}110_2^1(x_t^i)]$. Such rule is also, therefore, computationally irreducible.

4.3. Continuum limit

Although, as defined, Eqs. (11) and (13) apply to a finite set of integers $\in [0, p-1]$ which are mapped to the integers in the same interval, if $p \to \infty$ the above expression may reproduce with arbitrary precision any continuous map. As an example $l = r = 0$ is assumed in the following. The number of symbols p can now play the role of the "precision" in the sense that a real number is given in terms of multiples of $1/p$ and then a mapping of the rational numbers contained in the interval $[0, 1]$ to itself is performed. This can be seen from Eq. (11) since for rules ${}^0R^0_p$ one has

$$x^i_{t+1} = \sum_{n=0}^{p-1} a_n \mathcal{B}\left(n - x^i_t, \frac{1}{2}\right) = \sum_{n=0}^{p-1} a_n \mathcal{B}\left(\frac{n - x^i_t}{p-1}, \frac{1}{2(p-1)}\right) \tag{47}$$

where the property $\mathcal{B}(ax, \epsilon) = \mathcal{B}(x, \epsilon/a)$ for an integer value a (which can be proved directly from Definition 1) has been used. By defining the real quantities $n' = n/(p-1)$, $y^i_t = x^i_t/(p-1)$ and $y^i_{t+1} = x^i_{t+1}/(p-1)$, by taking the limit $p \to \infty$, and by using that $\lim_{\epsilon \to 0} \mathcal{B}(x, \epsilon) = 2\epsilon\delta(x)$ (where $\delta(x)$ denotes the Dirac delta function)

$$y^i_{t+1} = \lim_{p\to\infty} \int_0^1 dn' \frac{a_{n'p}}{(p-1)^2} \delta\left(n' - y^i_t\right) \tag{48}$$

The latter expression, involving real quantities constitutes the limiting behavior of a CA map involving an infinitely large number of integers. The logistic map, for example, can be reproduced if

$$a_{n'p} = r(p-1)^2 n'\left(1 - n'\right) \tag{49}$$

since in such case Eq. (48) becomes

$$y^i_{t+1} = r y^i_t\left(1 - y^i_t\right) \tag{50}$$

In terms of the original map involving only integers a CA rule ${}^0R^0_\infty$ described by Eq. (11) with

$$a_n = r(p-1)n\left(1 - \frac{n}{p-1}\right) \tag{51}$$

and where the code R is given by

$$R \equiv \sum_{n=0}^{p^{l+r+1}-1} a_n p^n = \lim_{p\to\infty} \sum_{n=0}^{p-1} rn(p-1)\left(1 - n/(p-1)\right)p^n \tag{52}$$

coincides with the logistic map in its dynamical behavior. Simulations show that if the limit is not taken and $p = 1000$, Eq. (47) (obtained from Eq. (11) in this case) already provides an excellent approximation to the logistic map.

In this concise treatment only rules of the form ${}^0R^0_\infty$ have been considered. If l or r are different to zero, differential operators (the Laplacian, for example) can also be reproduced. These operators are indeed a small subset of the infinitely many that can be constructed for each specific purpose and the $\mathcal{B}$-calculus introduced above can be thought as a formalism unifying discrete and continuum dynamics in an analogous way as how classical mechanics emerges from quantum mechanics in the limit of large quantum numbers.

5. Conclusions and perspectives

In this Letter a universal map encompassing all 1D deterministic first order in time CAs has been derived and generalizations to 2D and 3D, reversible CAs and higher-order-in-time 1D CAs have been introduced. These maps do not contain any freely adjustable parameters and are valid for any alphabet size and any neighborhood range (including non-symmetrical neighborhoods). For totalistic CAs the maps reduce to simpler expressions. Specific maps have been provided for the famous Conway's Game of Life and Wolfram's 256 elementary CAs. Many elementary CAs can be solved for the orbit from Eq. (11). An inductive method for CAs has been introduced for this purpose whose goal achieves with CA the same as integration does in continuum systems. Some other CAs are computationally irreducible, and it is not possible to find such expressions for the orbit. Still, in these latter cases, the mathematical expressions for the maps derived here give insight in how information propagates such that no shortcut is possible, as well as some deep relationships between different computationally irreducible CAs.

All CAs described in Wolfram's book [1] can thus be handled with mathematical expressions within the framework of the $\mathcal{B}$-calculus (i.e. the calculus with boxcar functions as introduced in Section 2). $\mathcal{B}$-calculus can also be used to deal with other rule-based dynamical systems as mobile cellular automata, Turing machines, substitution systems [1] and every other kind of finite automaton [32].

Standard mathematical methods, as symmetry analyses, can now be applied to CAs. Since each CA is associated to a particular simple program [1] this in turn implies that every algorithm can be expressed, in principle, by means of standard mathematical equations and that standard mathematics allows indeed to classify all different kinds of algorithms that can be envisaged. Such classification, in terms of symmetry considerations, will be published elsewhere [27]. Indeed, the formalism introduced here can be shown as well to yield a rigorous path to the origin of complexity in physical systems [27]. The methods presented here might also be useful in reaching conclusions on the important question "$P = NP$?" of the theory of computation [32].

Acknowledgements

Thanks to Katharina Krischer and Jose Manuel García Sanchis for fruitful conversations. Financial support from the Technische Universität München – Institute for Advanced Study, funded by the German Excellence Initiative, is also gratefully acknowledged.

References

[1] S. Wolfram, A New Kind of Science, Wolfram Media Inc., Champaign, IL, 2002.
[2] S. Wolfram, Cellular Automata and Complexity: Collected Papers, Addison-Wesley, Reading, MA, 1994.
[3] Physica (Amsterdam) D issues No. 10 and No. 45 are devoted to CA.
[4] E. Fredkin, Physica D 45 (1990) 254.
[5] S. Wolfram, Nature 311 (1984) 419.
[6] S. Wolfram, Rev. Mod. Phys. 55 (1983) 601.
[7] S. Wolfram, Physica D 10 (1984) 1.
[8] J. von Neumann, in: A.W. Burks (Ed.), Theory of Self-Reproducing Automata, University of Illinois Press, Champaign, IL, 1966.
[9] E.F. Codd, Cellular Automata, Academic Press, New York, 1968.
[10] S. Wolfram, Phys. Rev. Lett. 54 (1985) 735.
[11] S. Wolfram, Phys. Rev. Lett. 55 (1985) 449.
[12] A. Wuensche, M. Lesser, The Global Dynamics of Cellular Automata, Addison-Wesley, Reading, MA, 1992.
[13] A. Adamatzky, Identification of Cellular Automata, Taylor and Francis, London, 1994.
[14] L.O. Chua, A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science, vols. I–IV, World Scientific, Singapore, 2006–2011.
[15] C.G. Langton, Physica D 12 (1990) 42.
[16] W. Li, N.H. Packard, C.G. Langton, Physica D 45 (1990) 77.
[17] T. Toffoli, Physica D 10 (1984) 117.
[18] A. Ilachinski, Cellular Automata: A Discrete Universe, World Scientific, Singapore, 2001.
[19] D. Griffeath, C. Moore (Eds.), New Constructions in Cellular Automata, Oxford University Press, New York, 2003.
[20] T. Ceccherini-Silberstein, M. Coornaert, Cellular Automata and Groups, Springer Verlag, Heidelberg, 2010.
[21] H.V. McIntosh, One Dimensional Cellular Automata, Luniver Press, Frome, UK, 2009.
[22] O. Martin, A.M. Odlyzko, S. Wolfram, Comm. Math. Phys. 93 (1984) 219.
[23] E. Ahmed, A.S. Elgazzar, Physica A 373 (2007) 354.
[24] V. Garcia-Morales, K. Krischer, Contemp. Phys. 53 (2012) 79.
[25] R.G. Bartle, The Elements of Integration and Lebesgue Measure, Wiley, New York, 1966.
[26] M.J.D. Powell, Approximation Theory and Methods, Cambridge University Press, Cambridge, 1981, Chapter 4.
[27] See V. Garcia-Morales, arXiv:1203.3939.
[28] M. Gardner, Sci. Am. 223 (1970) 120.
[29] M. Gardner, Sci. Am. 224 (1971) 112.
[30] K. Dohmen, Improved Bonferroni Inequalities with Applications: Inequalities and Identities of Inclusion–Exclusion Type, Springer-Verlag, Berlin, Germany, 2003.
[31] R.L. Goldstein, Boolean Algebra, Dover, New York, 2007.
[32] M. Sipser, Introduction to the Theory of Computation, Course Technology, Boston, MA, 2006.