Asynchronous cellular automata

Nazim Fatès

Asynchronous cellular automata

Nazim Fatès
Inria Nancy – Grand Est
615 rue du Jardin Botanique, 54 600 Villers-lès-Nancy ; France,
nazim.fates@inria.fr

Nov. 2017

Foreword

This text has been proposed for the Encyclopedia of Complexity and Systems Science edited by Springer Nature and should appear in 2018. Please refer to the final online version for an up-to-date version (see http://www.springer.com/us/book/9780387758886).

Article Outline

This text is intended as an introduction to the topic of asynchronous cellular automata and is presented as a path. We start from the simple example of the Game of Life and examine what happens to this model when it is made asynchronous (Sec. 1). We then formulate our definitions and objectives to give a mathematical description of our topic (Sec. 2). Our journey starts with the examination of the shift rule with fully asynchronous updating and from this simple example, we will progressively explore more and more rules and gain insights on the behaviour of the simplest rules (Sec. 3). As we will meet some obstacles in having a full analytical description of the asynchronous behaviour of these rules, we will turn our attention to the descriptions offered by statistical physics, and more specifically to the phase transition phenomena that occur in a wide range of rules (Sec. 4). To finish this journey, we will discuss the various problems linked to the question of asynchrony (Sec. 5) and present some openings for the readers who wish to go further (Sec. 6).

Definition of the Subject

This article is mainly concerned with asynchronous cellular automata viewed as discrete dynamical systems. The question is to know, given a local rule, how the cellular automaton evolves if this local rule is applied to only a fraction of the cells. We are mainly interested in stochastic systems, that is, we consider that

the updating schemes, the functions which select the cells to be updated, are defined with random variables. Although there exists a wide range of results obtained with numerical simulations, we focus our discussion on the analytical approaches as we believe that the analytical results, although limited to a small class of rules, can serve as a basis for constructing a more general theory. Naturally, this is a partial view on the topic and there are other approaches to asynchronous cellular automata. In particular, such systems can be viewed as parallel models of computation ([[INTERNAL REF.]] See Th. Worsch's article in this encyclopedia) or as models of real-life systems. Readers who wish to extend their knowledge may refer to our survey paper for a wider scope on this topic [Fat14a].

1 Introduction

Cellular automata were invented by von Neumann and Ulam in the 1950's to study the problem of making artificial self-reproducing machines [Moo62]. In order to imitate the behaviour of living organisms, the design of such machines involved the use of a grid where the nodes, called the *cells*, would evolve according to a simple recursive rule. The model employs a unique rule, which is applied to all the cells simultaneously: this rule represents the "physics" of this abstract universe. The rule is said to be *local* in the sense that each cell can only see some subset of cells located at short distance: these cells form its *neighbourhood*. In von Neumann's original construction, all the cells are updated at each time step and this basis has been adopted in the great majority of the cellular automata constructions. This hypothesis of a perfect parallelism is quite practical as it facilitates the mathematical definition of the cellular automaton and its description with simple rules or tables. However, it is a matter of debate to know if such a hypothesis can be "realistic". The intervention of an external agent that updates all the components simultaneously somehow contradicts the locality of the model. One may legitimately raise what we could call the *no-chief-conductor* objection: "in Nature, there is no global clock to synchronise the transitions of the elements that compose a system, why should there be one in our models?".

However, this objection alone cannot discard the validity of the classical synchronous models. Instead, one may simply affirm that the no-chief-conductor objection raises the question of the *robustness* of cellular automata models. Indeed, at some point, the hypothesis of perfectly synchronous transitions may seem unrealistic but we cannot know *a priori* if its use introduces spurious effects. There are some cases where a given behaviour of a cellular automaton will only be seen for the synchronous case and there are also cases where this behaviour remains constant when the updating scheme is changed. In essence, without any information on the system, we have no means to tell what are the consequences of choosing one updating scheme or the other.

If we have a *robust* model, changes in the updating may only perturb slightly the global behaviour of a system. On the contrary, if this modification induces

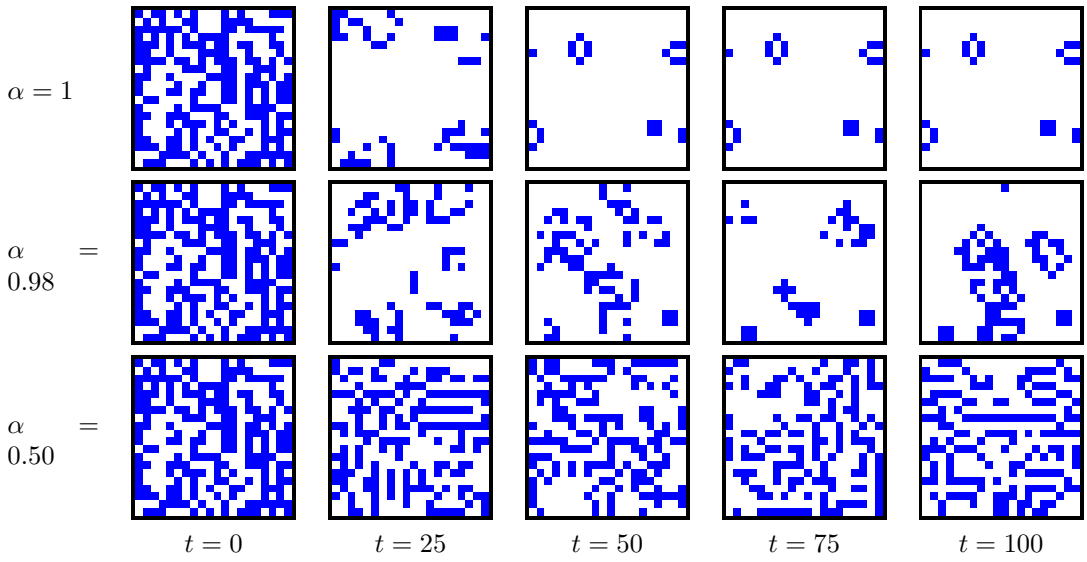


Figure 1: Configurations obtained with the α-asynchronous Game of Life for three values of the synchrony rate α and the same initial conditions. (top): synchronous updating, the system is stable at $t = 50$; (middle): small asynchrony introduced, the system is still evolving at $t = 100$; (bottom): $\alpha = 1/2$, the qualitative behaviour of the system has changed.

a qualitative change on the dynamics, the model will be called *structurally unstable* or simply *sensitive* to the perturbations of its updating scheme. A central question about cellular automata is thus to know how to assess their degree of robustness to the perturbations of their updating. Naturally, the same questions can be raised about the other hypotheses of the model: the homogeneity of the local rule, the regular topology, the discreteness of states, etc. (see e.g. Problem 11 in Ref. [Wol85]).

1.1 A first experiment

In order to make things more concrete, we propose to start our examination with a simple asynchronous CA. We will employ the *α-asynchronous* updating scheme [FM05] and apply the following rule: at each time step, each cell is updated with a probability α and is left in the same state with probability $1-\alpha$. The parameter α is called the *synchrony rate* (see the formal definitions below)[1]. The advantage of this definition is to control the robustness of the model by varying the synchrony rate continuously from the classical synchronous case $\alpha = 1$ to a small value of α, where most updates will occur sequentially. We thus propose to examine the behaviour of the α-asynchronous Game of Life.

[1]Note that from the point of view of a given cell, all happens as if between two updates each cell was waiting a random time that follows a geometric law of parameter α.

Figure 1 shows three different evolutions of the rule: the synchronous case ($\alpha = 1$), an evolution with a little asynchrony ($\alpha = 0.98$) and an evolution with a stronger asynchrony ($\alpha = 0.5$).

The first observation is that the introduction of a small degree of asynchrony does not modify the *qualitative* behaviour of the rule on the short term. However, one can predict that the long-term behaviour of the rule *will* be perturbed because it is no longer possible to observe cycles. For example, the configuration with only three living cells in a row oscillates in the classical Game of life, but these oscillations only exist with a synchronous updating and the configuration evolves to a totally different pattern when this perfect simultaneity is broken. Another important property to remark is that the new (asynchronous) system has the same fixed points as the original (synchronous) system. In fact, this is a quite general property that does not depend on the local rule. The reason is simple: if a configuration is a fixed point of the synchronous system, it means that all its cells are stable under the application of the local rule. Hence, if we select a subset of cells for an update, this subset will also be stable. Reciprocally, if any choice of cells gives a stable situation, then the whole system is also stable.

The second important observation regards the evolution with $\alpha = 0.5$: the global behaviour of the system is completely overwhelmed! A new stationary behaviour appears, and a pattern which resembles a labyrinth forms. This pattern is stable in some parts and unstable in some other parts of the grid. We will not enter here into the details on how this stability can be quantified but it is sufficient to observe that, in most cases, one observes that this pattern remains a very long time.

1.2 Questions

It may be argued that these observations are not that surprising, because if one modifies the basic definitions of a dynamical system, one naturally expects to see effects on its behaviour. However, this statement is only partially true, as this type of radical modifications is not observed for *all* the rules. In fact, as Nakamura has shown, we can always modify a rule in order to make it insensitive to the variations of its updating scheme [Nak74, Nak81]. Formally, this amounts to show that any classical deterministic cellular automaton may be simulated by an asynchronous one. By "simulated" we mean that the knowledge of the evolution of the stochastic asynchronous system allows one to know the evolution of the deterministic original rule with a simple transformation ([[INTERNAL REF.]] See Th. Worsch's article). The idea of Nakamura is that each cell should keep three registers: one with its current state, one with its previous state, and one with a counter that tells if its local time is late, in advance or synchronised with the local time of its neighbours. There is of course an overhead in terms of simulation time and number of states which are used, and one may want to reduce this overhead as much as possible [LAPM04], but the point is that there are asynchronous rules which will evolve as their synchronous deterministic counterparts. As an extreme example, we can also think of the rule where each cell turns to a given state independently of its neighbour: the global evolution

is easily predicted.

Partial robustness can also be observed with some simple rules. For example, let us consider the majority rule: cells take the state that is the most present in their neighbourhood. Observing this rule on a two-dimensional grid with periodic boundary conditions shows that it is robust to the variations of α: roughly speaking, if we start from a uniform random initial condition, for $0.5 < \alpha < 1$, the system seems to always stabilise quickly on a fixed point. For smaller values of α the only noticeable effect is a slowdown of the converge time. However, a modification also exists at the vicinity of $\alpha = 1$: like for the Game of Life, as soon as a little asynchrony is present, cycles disappear.

These experiments indicate that there is something about asynchronous systems that deserves to be investigated. Since the first numerical simulations [BI84], a great number of approaches have been adopted to gain insights on asynchronous cellular automata. However, if we want to be convinced that these systems can be studied and understood theoretically, and despite their randomness, we need some analytical tools. The purpose of the lines that follow is to give a few indications on how the question of asynchrony in cellular automata can be dealt with with theoretical tools from computer science and probability theory.

2 Defining asynchrony in the cellular models

Literally, a-syn-chronous is a word derived from the Ancient Greek *ἀσυνχρονος*, which simply means: not-same-time. From this etymology, it follows that we cannot speak of *a single* model of asynchrony in cellular automata but there is an infinity of models. In fact, one is allowed to speak of an asynchronous model as soon as there is some perturbation in the updating process of the cells[2].We voluntarily stay vague at this point in order to stress that one may imagine a great variety of situations where some irregularity occurs on the way the information is processed by the cells. For instance, we may examine what happens if all the transitions *do* occur at each time step but where the cells receive the state of their neighbours imperfectly.

In this text, we will restrict our scope to the most simple cases of asynchronous updating.

2.1 Mathematical framework

Let $\mathcal{L} \in \mathbb{Z}^d$ be the set of cells that compose a d-dimensional cellular automaton. The set of states that each cell may hold is Q. The collection of all states at given time is called a *configuration* and the configuration space is thus $Q^{\mathcal{L}}$.

Let $\mathcal{N} \in (\mathbb{Z}^d)^k$ be the neighbourhood of the cellular automaton, that is, for $\mathcal{N} = (n_1, \ldots, n_k)$, n_i represents the vector between the central cell and its i-th

[2]Note that *asynchrony* and *asynchronism* have been both used in the literature in an equivalent way. We will in general use the former for the modification of the updating and use the latter to designate a topic of research.

neighbour.

The local function of a cellular automaton is a function $f : Q^k \to Q$ which assigns to a cell $c \in \mathcal{L}$ its new state $q' = f(q_1, \ldots, q_k)$, where the t-uple $(q_1, \ldots, q_k)$ represents the state of the neighbours of a cell c.

Starting from an initial configuration $x \in Q^{\mathcal{L}}$, the classical evolution of the system gives a sequence of configurations that we denote by $(x^t)_{t\in\mathbb{N}}$. This sequence is obtained by the recursive application of the global rule $F : Q^{\mathcal{L}} \to Q^{\mathcal{L}}$ defined with $x^0 = x$ and $x^{t+1} = F(x^t)$ such that:

$$\forall c \in \mathbb{Z},\ x_c^{t+1} = f\big(x^t_{c+n_1}, \ldots, x^t_{c+n_k}\big).$$

Now, to define an *asynchronous cellular automaton*, we need to introduce an *updating scheme*. Such a function takes the form $\mathcal{U} : \mathcal{L} \to \mathcal{P}(\mathcal{L})$, where $\mathcal{P}(S)$ denotes the parts of S, that is, the set of all subsets of S (also denoted by 2^S). For a given time step $t \in \mathbb{N}$, the set of cells that are updated at time t is represented by $\mathcal{U}(t)$.

We obtain a new global rule, denoted by $F_{\mathcal{U}} : \mathbb{N} \times Q^{\mathcal{L}} \to Q^{\mathcal{L}}$ where $F_{\mathcal{U}}(x, t)$ represent the image of x at time t given the updating scheme $\mathcal{U}$. The evolution of $(x^t)_{t\in\mathbb{N}}$ starting from $x \in Q^{\mathcal{L}}$ is now defined with $x^0 = x$ and $x^{t+1} = F_{\mathcal{U}}(x^t)$ such that:

$$\forall c \in \mathbb{Z},\ x_c^{t+1} = \begin{cases} f\big(x^t_{c+n_1}, \ldots, x^t_{c+n_k}\big) & \text{if } c \in \mathcal{U}(t), \\ x_c^t & \text{otherwise.} \end{cases}$$

The type of function $\mathcal{U}$ defines the type of asynchronism in use. The first issue of distinction is between *deterministic* and *stochastic* (or probabilistic) functions. In this text we will focus on stochastic functions. Indeed, since asynchronism is often thought of an unpredictable aspect of the system, stochastic systems have been more intensively studied. One finds only a small number of studies which use deterministic systems. Examples of such studies can be found in Ref. [SdR99, CGN05] where the authors have considered for example the effects caused by updating cells sequentially from left to right. As one may expect, such approaches often lead to curious phenomena: the information spreads in a non-natural way because a single sequence of updates from left to right suffices to change the state of the whole system. More interesting are *even-odd* updating schemes where one updates the even cells and, in the next step, the odd cells. A famous example of such model is the Q2R model [Vic84]: although the local rule of this system is deterministic, using a random initial condition makes it evolve with the same density as the Ising model (see e.g. Ref. [KT15] for a recent development).

In fact, we can remark that in general it is not difficult to transform an asynchronous system into a synchronous one: in many cases, adding more states is sufficient. For example, for the even-odd updating, we may mark the even and odd cells with a flag up and down, respectively, and make this flag "flip" at each time step. Similarly, an ordered updating may be simulated in a synchronous model by moving a token in a given order. However such direct transformations are not always possible: for example, Vielhaber has proposed an original way of

achieving computation universality by selecting the cells to update [Vie13] and this construction cannot be transformed into a deterministic cellular automaton by the mere addition of a few internal states.

2.2 Randomness in the updating

In the case where the updating scheme $\mathcal{U}$ is a random variable, then the evolution of the system is a *stochastic process*, and, if $\mathcal{U}$ does not depend on time, it is a Markov chain (a memoryless system). In order to be perfectly rigorous in the formal description of the system, advanced tools from probability theory are necessary. A good example on how to properly use these mathematical objects and their properties can be found in a survey by Marcovici and Mairesse [MM14]. However, for the sake of simplicity, one may still use the usual notations and consider that the sequences $(x^t)_{t\in\mathbb{N}}$ are formed by configurations rather than probability distributions.

We can now define the two major asynchronous updating schemes:

• α-asynchronous updating scheme: Let $\alpha \in (0, 1]$ be constant called the synchrony rate. Let $(\mathcal{B}_i^t)_{i\in\mathcal{L},t\in\mathbb{N}}$ be a sequence of independent and identically distributed Bernoulli random variables of parameter α. The evolution of the system with an α-asynchronous updating scheme is then given by:

$$x^0 = x \text{ and } \forall i \in \mathbb{Z},\ x_i^{t+1} = \begin{cases} f\left(x^t_{i+n_1}, \dots, x^t_{i+n_k}\right) & \text{if } \mathcal{B}_i^t = 1, \\ x_c^t & \text{otherwise.} \end{cases}$$

• *fully asynchronous* updating scheme: In the case where $\mathcal{L}$ is finite, let $(S_t)_{t\in\mathbb{N}}$ be a sequence of independent and identically distributed random variables that select an element uniformly in $\mathcal{L}$. The evolution of the system is given by:

$$x^0 = x \text{ and } \forall i \in \mathbb{Z},\ x_i^{t+1} = \begin{cases} f\left(x^t_{i+n_1}, \dots, x^t_{i+n_k}\right) & \text{if } i = S_t, \\ x_c^t & \text{otherwise.} \end{cases}$$

Note that in most cases, authors do not use the indices i and t for $(\mathcal{B}_i^t)$ or (S_t) and simply consider that there is *one* function that used at each time step and for each cell.

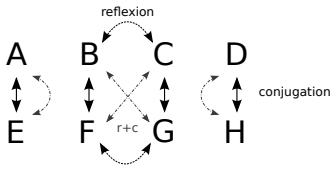
We do not enter here into the details of how we can generalise these definitions (see e.g. Ref. [DFMM13]). We point the work of Bouré et al. on asynchronous *lattice-gas* cellular automata to underline that adding asynchrony to the cellular models which have more structure than the classical ones can be a non-trivial operation if one wants to maintain the properties of these models (e.g. conservation of the number of particles) [BFC13a]. Similar difficulties arise when agents can move on the cellular grid and one needs to define some procedures to solve the conflicts that may occur when several agents want to modify simultaneously the same cell [CF10, BF12].

3 Convergence properties of simple binary rules

We have seen that a central question in the study of asynchronous cellular automata was to determine their convergence properties. In particular one may

Table 1: Notation by transitions. left: table of transitions and their associated labels. right: symmetries of the ECA space (see text for explanations)

A	B	C	D
000	001	100	101
010	011	110	111
E	F	G	H



wonder, given a simple binary rule, what we can predict about its possible behaviour. Is it converging to a given fixed point? In which time in average? And if so, what kind of "trajectory" the system will follow to attain a stable state (if any)? The lines that follow aim at presenting the mathematical tools to answer these questions.

3.1 Expected convergence time to a fixed point

Recall that one major modification caused by the transformation of a cellular automaton from synchronous to asynchronous updating is the removal of cycles: cycles are replaced by some attractive sets of configurations (see below for a more precise description). Let us examine this property on a simple case. We work on a finite one-dimensional system and denote the set of cells by $\mathcal{L} = \mathbb{Z}/n\mathbb{Z}$, where n is the number of cells. We employ a fully asynchronous updating scheme described by a sequence of independent and identically distributed random variables (S_t) which are uniform on $\mathcal{L}$ (one cell is selected at each time step). The local rule depends only on the state of the cell itself and its left and right neighbours, we have: $\mathcal{N} = \{-1, 0, 1\}$. Recall that for an initial condition $x \in Q^{\mathcal{L}}$, the evolution of the system is thus described by $(x^t)_{t\in\mathbb{N}}$ such that: $x^0 = x$ and $x^{t+1} = F(x^t)$ such that $\forall i \in \mathcal{L}$, $x_i^{t+1} = f(x_{i-1}^t, x_i^t, x_{i+1}^t)$ if $i = S_t$ and $x_i^{t+1} = x_i^t$ otherwise.

To evaluate the converge time of given rule, we proceed as in the theory of computation and define the "time complexity" of this rule as the function which estimates the amount of time taken by the "most expensive" computation of size n [(see Th. Worsch's article in this encyclopedia)]. Let $\mathcal{F}$ denote the set of fixed points of a rule and let $T(x)$ denote the random variable which represents the time needed to attain a fixed point: $T(x) = \{\min t : x^t \in \mathcal{F}\}$. In order to have a fair comparison with the synchronous update, we consider that *one time step* corresponds to n updates and we introduce the *rescaled time* $\tau(x) = T(x)/n$. The "complexity measure" of a rule is than given by the *worst expected convergence time* (WECT): $WECT(n) = \max\left\{\mathbb{E}\{\tau(x)\}; x \in Q^{\mathcal{L}}\right\}$.

3.2 Two notations for ECAs

Following a convention introduced by Wolfram, it is common to identify each ECA f with a decimal code $W(f)$ which consists in converting the sequence

of bits formed by the values of f to a decimal number: $W(f) = f(0,0,0).2^0 + f(0,0,1).2^1 + \cdots + f(1,1,1).2^7$. We now introduce another notation of ECA rules, which consists in identifying an ECA rule f with a word which consists in a collection of labels from $\{\mathsf{A}, \mathsf{B}, \ldots, \mathsf{H}\}$ where each label identifies an *active* transition, that is, a couple $\big((x,y,z), f(x,y,z)\big)$ such that $f(x,y,z) \neq y$. The mapping between labels and transition is given in Table 1.

For example, let us consider the XOR rule $f(x,y,z) = x \oplus y \oplus z$, where $\oplus$ denotes the usual XOR operator. The decimal code associated to this rule is 150. The active transitions of this rule are $\mathtt{001} \to \mathtt{1}$ (B), $\mathtt{100} \to \mathtt{1}$ (C), $\mathtt{011} \to \mathtt{0}$ (F) and $\mathtt{110} \to \mathtt{0}$ (G). The four other transitions are passive, that is, they do not change the state of the central cell. We thus obtain the new code: BCFG.

Knowing the transition code of a rule, one can easily deduce the symmetric rules: to obtain the rule where the left and right directions are permuted, it is sufficient to exchange the letters B and C and to obtain the symmetric rule where the states $\mathtt{0}$ and $\mathtt{1}$ are permuted, on exchanges the letters A and E, B and F, C and G and H (see Tab. 2-right)

In the case of a fully asynchronous updating, the notation by transitions also allows us to decompose the behaviour of the local rule as follows:

- If a rule does not have A (resp. H) in its code, the size of a $\mathtt{0}$-region (resp. a $\mathtt{1}$-region) may increase or decrease by 1 but this region cannot be broken.
- Transitions B and F control the movements of the $\mathtt{01}$-frontiers: B (resp. F) moves this frontier to the left (resp. to the right). If both transitions are present, the $\mathtt{01}$-frontier performs a non-biased random walk.
- Similarly, transitions C and G control the movements of the $\mathtt{10}$-frontiers.
- Transition D (resp. E) controls the fusion of $\mathtt{1}$-regions (resp. $\mathtt{0}$-regions): the absence of D (resp. E) implies that the $\mathtt{0}$-regions (resp. $\mathtt{1}$-regions) cannot disappear.

These properties are summed up on Tab. 2-left.

In addition, the code by transitions can be used to produce a complementary useful view on configurations by transforming a configuration $x \in Q^{\mathcal{L}}$ in a configuration $\tilde{x} \in \{\mathsf{a}, \ldots, \mathsf{h}\}^{\mathcal{L}}$, where each cell is labelled with a, b, ... if the transition A, B, ... applies on it. An example of such transformation is shown on Fig. 2-left. This transformation can be done directly but it also interesting to consider the *de Bruijn graph* (or diagram), which allows one to do this transformation by reading one symbol at time, from left to right, and by following the edge with the label that was read (see Fig. 2-right). This graph is useful for determining various properties of cellular automata. For example, the fixed points of rule are made by the cycles which do not contain a node with an active transition. For any configuration x, if we write by a, b, ... the respective number of a's, b's, ... of $\tilde{x}$, then the following relations can be easily obtained: $\mathsf{b} = \mathsf{c}$; $\mathsf{f} = \mathsf{g}$; $|x|_{\mathtt{01}} = \mathsf{b} + \mathsf{d} = \mathsf{e} + \mathsf{f}$; $|x|_{\mathtt{10}} = c + d = e + g$; $|x|_{\mathtt{01}} = |x|_{\mathtt{10}}$.

Table 2: left : Summary of the effect of each transition on a fully asynchronous ECA. right: Summary of the combinations of two (active or inactive) transitions.

A	stability of 0-regions
B	01-frontiers move left
C	10-frontiers move right
D	absorption of 0-regions
E	absorption of 1-regions
F	01-frontiers move right
G	10-frontiers move left
H	stability of 1-regions

no A+ no H	doubly quiescent rule
B+ F	random walk of the 01-frontiers
C+ G	random walk of the 10-frontiers

```
0011001111100
abfgcbfhhhgca
      ↑
0011001011100
abfgcbedfhgca
```

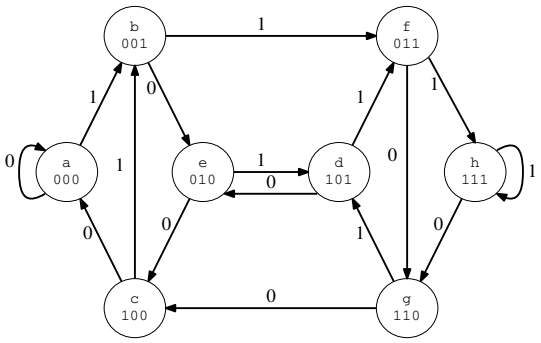


Figure 2: (left): Example of two binary configurations and their images by the transition code. The upper configuration is obtained by updating the lower configuration on the cell indicated with an arrow. (right): De Bruijn graph with the correspondence between binary sequences of length 3 and transitions A, . . . , H. The label on the edges show the next letter that in input when reading a binary sequence from left to right.

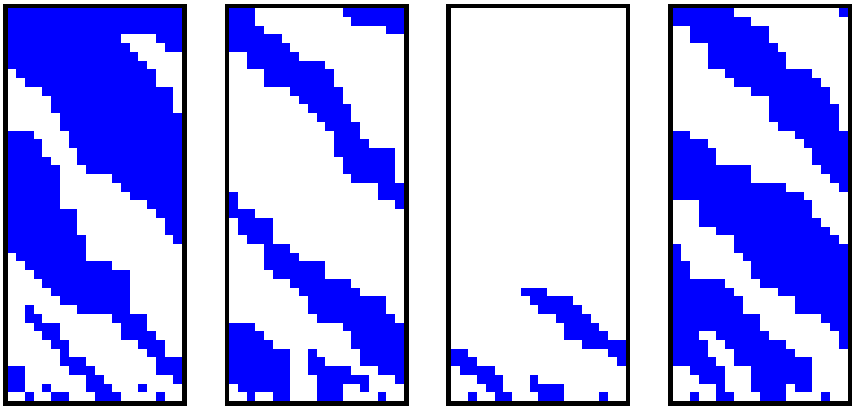


Figure 3: Space-time diagrams showing the evolution of the shift rule for a ring of n cells, with $n = 20$. Cells in blue and white respectively represent states 0 and 1. Time goes from bottom to top. Each row shows the state of the system after n random updates. This convention is kept in the following.

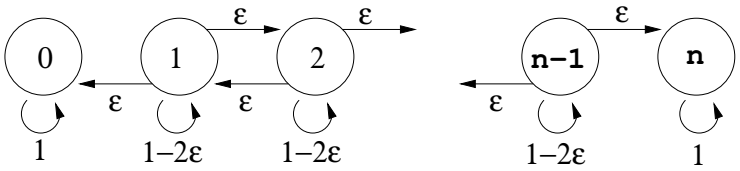


Figure 4: Representation of the Markov chain that counts the number of 1's. The constant $\epsilon = 1/n$ represents the probability to update a cell at a given time step.

3.2.1 A starting example

Let us take the shift rule $f(x, y, z) = z$ as a first example of ECA. The wolfram code and the transition code of this rule is 150:BCFG. As it can be seen from the space-time diagrams shown on Fig. 3, although the local rule is elementary, the evolution of the system is quite puzzling at first sight. The diagrams show that, starting from the *same* initial condition, the system may reach either the fixed point $\mathbf{0} = 0^{\mathcal{L}}$ or the fixed point $\mathbf{1} = 1^{\mathcal{L}}$, and that the convergence time is subject to a high variability. A little close-up on the behaviour of the rule allows us to discover that the number of regions of 0's or 1's can only decrease. Indeed, it is impossible to create a new state inside a region of homogeneous state. More precisely, a change of state can only occur on the boundaries between regions: if such a boundary is updated, it moves one cell to the left.

Let us examine what happens for an initial condition $x \in Q^{\mathcal{L}}$ with only two regions: we have $|x|_{01} = 1$, where $|x|_{01}$ is the function which counts the number of occurrences of the pattern 01 in x. To calculate the probability to reach a

given fixed point, we introduce the stochastic process (X_t) which counts the number of 1's of a configuration: $X_t = |x^t|_1$. where $|x|_1$ is the function which counts the number of occurrences of 1's. It can easily be verified that (X_t) *is* a Markov chain whose graph is shown on Fig. 4. Note that this is not immediate and this property is not true for any initial condition. The values $X_t = 0$ and $X_t = n$ are the absorbing states of the Markov chain and represent the convergence of the asynchronous cellular automaton to its respective fixed points $\mathbf{0}$ and $\mathbf{1}$. We can thus calculate the probability $p_{\mathbf{1}}(k)$ to reach the fixed point $\mathbf{1}$ given an initial condition x such that $|x|_1 = k$. This can be done by recurrence by noting that $p(0) = 0$, $p(n) = 1$ and $p(k) = \epsilon p(k-1) + (1-2\epsilon)p(k) + \epsilon p(k+1)$, where $\epsilon = 1/n$ is the probability to update a cell. The solution is $p(i) = \epsilon i = i/n$. In words, the probability to reach the fixed point $\mathbf{1}$ is exactly the density of the initial configuration.

Let us now estimate the average number of time steps that it will take to reach one of the two fixed points. Recall that: $T_(x) = \min\{t \in \mathbb{N} : x^t \in \{\mathbf{0}, \mathbf{1}\}\}$. As the Markov chain is finite and has two absorbing states, T is almost surely finite. The average of T depends only on the number of 1's of the initial condition. With a small abuse of notation, we can denote by T_i the average convergence time from an initial condition with i cells in state 1, we have the following recurrence equation: $T_0 = T_n = 0$ and $\forall i \in \{1, \ldots, n-1\}$,

$$\begin{aligned} T_i = &\quad \varepsilon(T_{i-1} + 1) + (1 - 2\varepsilon)(T_i + 1) + \varepsilon(T_{i-1} + 1) && (1) \\ = &\quad 1 + \varepsilon T_{i-1} - (1 - 2\varepsilon)T_i + \varepsilon T_{i+1} && (2) \end{aligned}$$

The solution of this system is $T_i = i(n-i)/2\varepsilon$. Since $\varepsilon = 1/n$, we can write $\forall i \in \{0, \ldots, n\}$, $T_i \leq n^3/8$; in other words, for the configurations with only two zones, the average number of updates needed to attain a fixed point is at most cubic in n.

Martingales

How can we deal with the other configurations? If we start from a configuration x with k 1-regions and $k > 1$, the probability to increase or decrease by 1 the number of 1's is $k\epsilon$. The evolution of the system can no longer be described by the Markov chain of Fig. 4. Indeed, the value ϵ needs to be replaced by $\epsilon' = k\epsilon$ but, as k is not constant, this process is no longer a Markov chain. As seen on Fig. 4, the frontiers of the regions will perform random walks until a region disappears, which will make ϵ' decrease again, and so on until we reach one of the two fixed points. In order to determine the convergence time $\tau(x)$, one could estimate the average "living time" of a configuration with k-regions. However, this is a difficult problem because this living time strongly depends on the size of each region.

It is easier to note that the process (X_t) defined with $X_t = |x^t|_1$ is a *martingale*, that is, a stochastic process whose average value is constant over time. The theory of martingales allows us to find the probability $p_{\mathbf{1}}(x)$ to reach the fixed point $\mathbf{1}$ from x and the average time of convergence $\mathbb{E}\{\tau(x)\}$. For

the sake of brevity, we skip the details of the mathematical treatments and write down directly the results that are exposed in Ref. [FMST06]: (a) the probability of reaching the fixed point $\mathbf{1}$ is still equal to the initial density: $p_{\mathbf{1}}(x) = |x|_1/n$, (b) the rescaled average time also scales quadratically with n: $\mathbb{E}\{\tau(x)\} \leq |x|_1(n - |x|_1)/2$.

We thus have an upper bound on the WECT which is $WECT(n) \leq n^2/8$ and, considering the initial condition $x = \mathtt{0}^{n/2}\mathtt{1}^{n/2}$, we obtain the lower bound: $WECT(n) \geq n^2/8$. We can thus write $WECT(n) = \Theta(n^2)$ where Θ expresses the equivalence up to a constant. We thus say that the shift has a *quadratic* convergence time or, for short, that it is quadratic.

A relationship with computational problems

In fact, since the convergence of the asynchronous shift depends on the initial density, one may consider this process as a particular kind of decentralised *computation*. For the sake of brevity, we will not develop this point here but we simply indicate to the readers interested by this issue that similar stochastic rules have been used to solve the density classification problem (see e.g. Ref. [Fuk02, Fat13b, dO14] and de Oliveira's article in this encyclopedia).

3.3 From the shift to other quadratic rules

We now examine step by step how to generalise the example of the asynchronous shift given above to a wider class of rules.

With the decomposition described above, we can readily deduce that the Markov chain described for counting the number of 1's for the shift rule (BDEG) also applies for rule CG, for which the 10-frontier performs a non-biased random walk and for rule BCDEFG, for which the two frontiers perform a random walk.

In a second time, we can ask what happens if we change the code of these rules by removing the transition D of their code, that is, we set $101 \to 0$ and make the transition D inactive. This transformation implies that the 0-regions can no longer disappear while the 1-regions may disappear if an isolated 1 is updated ($010 \to 0$). As a consequence, the fixed point $\mathbf{1}$ is no longer reachable and the system will almost surely converge to the fixed point $\mathbf{0}$ for an initial condition different from $\mathbf{1}$. The system will thus most of the time behave as a regular martingale but sometimes it will "bounce" on an isolated 0. Is the average convergence time still quadratic? The answer is positive: even though the behaviour cannot be described (simply) by a martingale, it is possible to "save" the previous results and still obtain a quadratic scaling of the WECT. Interested readers may refer to our study on fully asynchronous doubly quiescent[3] rules for the mathematical details [FMST06].

[3] A *quiescent state* is a state q such that the local rule f obeys $f(q, \ldots, q) = q$.

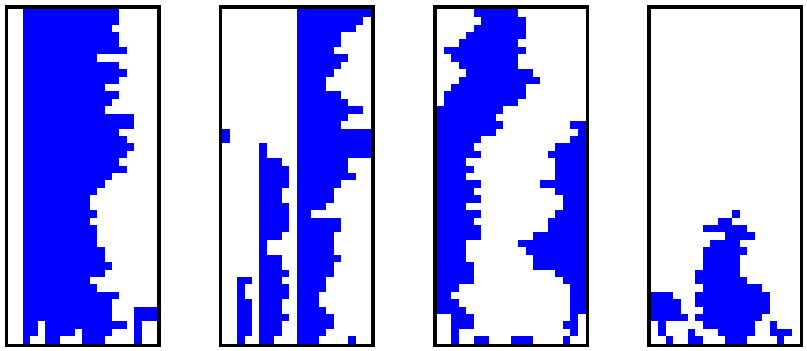


Figure 5: Space-time diagrams showing the evolution of four rules with a quadratic worse expected convergence time (WECT).

3.4 Functions with a potential

In the previous paragraph, we started from the shift rule (BDEG), showed that it had a quadratic WECT, an then indicated that five other rules had a similar qualitative behaviour and a quadratic WECT. The other rules were obtained by making the transition D inactive or by changing the behaviour of the frontiers, as long as this movement remained a non-biased random walk.

We now propose to examine what happens if we dare to "touch" a transition that breaks the random movement of the frontiers. Concretely, let us make the transition B inactive: we obtain the minimal representative rule DEG (168). The evolution of this rule is displayed on Fig. 6; it can be seen that the evolution of the rule is less "spectacular" than the quadratic rules. The size of the 1-regions regularly decrease until the all the regions disappear and the system reaches the fixed point $\mathbf{0}$. It is easy to see that in the case where the initial condition does not contain an isolated 0, the evolution of the number of 1's is a non-increasing function. Now, let us consider the function $\phi : Q^{\mathcal{L}} \to \mathbb{N}$ defined by $\phi(x) = |x|_1 + |x|_{01}$. Writing $(X_t) = \phi(x^t)$, one can verify that the evolution of (X_t) is non-increasing. Indeed, if a transition D is applied, the number of 1's increases by 1 but the number of regions also decreases by 1. Moreover, we have that $X_t = 0$ implies that $x^t = \mathbf{0}$. The function ϕ can thus be named *a potential*: it is a positive, non-increasing function of the current state of the system, which equals zero when the system has attained its attractive fixed point. This argument can be applied for showing a linear WECT for the following four rules (G is active): 136:EG, 140:G, 168:DEG, 172:DG. and the following four rules (F and G are active): 128:EFG, 132:FG, 160:DEFG, 164:DFG.

Interestingly, a similar type of convergence can also be obtained by *adding* an active transition to the shift rule. For example, let us consider ECA BDEFG

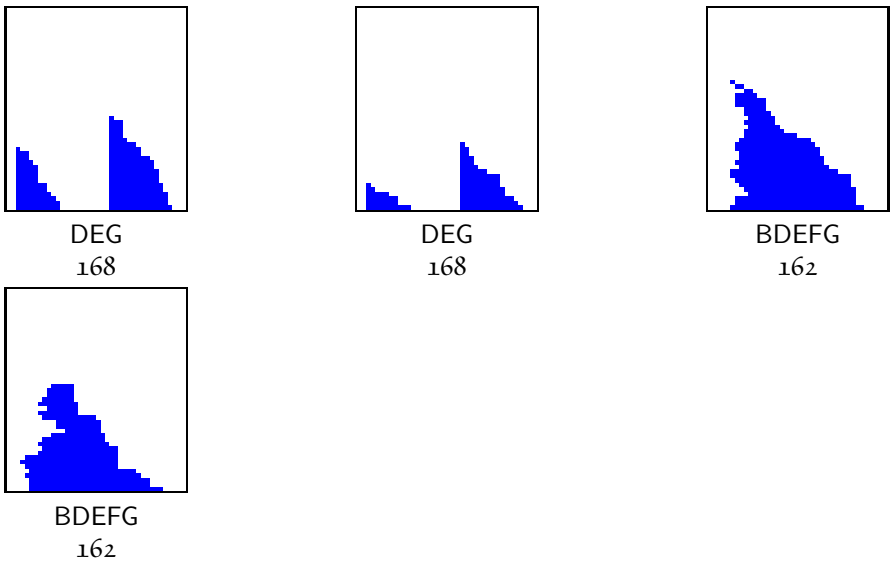


Figure 6: Space-time diagrams showing two evolutions of two rules with a linear worse expected convergence time (WECT).

(162). Its evolution is shown on Fig. 6. One should observe that the 01-frontiers perform a non-biased random walk while the 10-frontier tends to move to the left. This means that the 1-regions have a tendency to decrease, but their evolution is no longer monotonous as in the case of rule DEG. It can be shown that if we take back the function $\phi(x) = |x|_1 + |x|_{01}$ and $X_t = \phi(x^t)$, then (X_t) is a super-martingale, that is, its average value decreases in average. This property and other conditions ensuring that it cannot stay too "static" imply that its convergence time which scales linearly with the ring size n [FMST06]. Indeed, for any configuration that is not a fixed point the quantity $\mathbb{E}\{X^{t+1} - X^t | x^t\}$ is negative. The same method can be applied for showing the convergence in linear time for the rule 130:BEFG.

Non-polynomial types of convergence

For the sake of brevity, we will not go here into the details but only indicate the other classes of convergence that were exhibited. Readers may consult Ref. [FMST06] for detailed arguments.

- The rules 200:E and 232:DE have a logarithmic WECT. This can be shown with the same techniques as for the convergence of the coupon-collector process [FMST06].
- The rule 154:BCEG has an exponential WECT. This comes from a kind of paradox: the rule has a tendency to increase the number of 1's but its

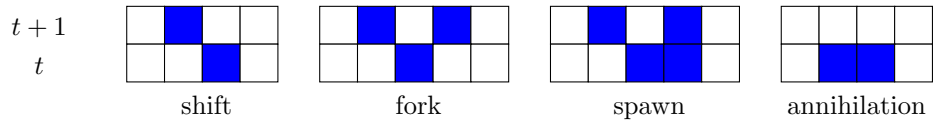


Figure 7: New phenomena observed with the α-asynchronous updating of linear CA (from the work of Regnault et al. [FRST06]).

only fixed point **1** is not reachable. The only way it can converge is by reaching the fixed point **0**, a phenomenon that is very unlikely.

- The rules 134:BFG, 142:BG, 156:CG and 150:BCFG are non-converging. This is because in all these rules transitions D and E are inactive and, at the same time, the frontiers are not static.

Other elementary rules

The question of classifying the other ECA rules, where no state or only one state is quiescent, is still open. Some conjectures have been stated from experimental observations but they still deserve an in-depth analysis [Fat13a]. In particular, there are currently only partial results for all the rules which are conjectured to converge "very rapidly", that is, in logarithmic time [Fat14b].

3.5 From fully asynchronous to α-asynchronous updating

What happens if one uses a partially synchronous updating scheme instead of a totally asynchronous one? Regnault et al. have extended the convergence results of the doubly-quiescent ECA to the case of α-asynchronous updating [FRST06]. The possibility of having simultaneous updates of neighbouring cells creates additional "local movements" and the behaviour of these rules is more difficult to analyse. In particular, the authors have identified four phenomena that are specifically related to the α-asynchronous updating: the shift, the fork, the spawn, the annihilation. These phenomena are shown on Fig. 7.

The authors developed an interesting analytical framework (potential functions, masks, etc.) and succeeded in giving bounds on the convergence of 19 (minimal) doubly-quiescent rules, leaving the question open four five other rules. the various rules show different kind of scaling relations of the WECT, depending on α and n. If we consider the dependence on n only, the families of functions are the same as those obtained for fully asynchronous dynamics, that is, logarithmic, linear, quadratic, exponential and infinite. However, there are rules whose type of converge varies from the fully asynchronous updating to α-asynchronous updating. For example, rule 152:CEG, which is quadratic with a fully asynchronous updating (see above), becomes linear for α-asynchronous updating. Two rules, namely ECA 146:BCEFG and 178:BCDEFG were conjectured to display a phase transition: their type of converge may change from polynomial to exponential depending on whether the α is greater or lower than

a particular critical value. This property was partially proved by Regnault in a thorough study of ECA 178, were the polynomial and exponential convergence times were formally obtained for extremal values of the synchrony rate [Reg13]. Ramos and Leite recently studied a generalisation of this model where the asynchronous case appears as a special case of the family of probabilistic cellular automata that are studied [RL17].

3.6 Two-dimensional rules

The study of the convergence properties of simple two-dimensional rules has been carried out for the so-called *totalistic cellular automata*, where the local rule only depends on the number of `1`'s in the neighbourhood [FG09]. For the von Neumann neighbourhood (the cell and its four nearest neighbours), there are 2^6 such rules. Their WECT were also analysed for the *fully asynchronous* updating and all rules but one were found to fall into the previous classes of convergence. One remarkable exception was given by the *epidemic rule*, where a `0` turns into a `1` if it has a `1` in its neighbourhood and then always remain a `1`. This rule has a WECT which scales as $\Theta(\sqrt{n})$. Even though this scaling property can be intuitively understood from the dynamics of the rule, which merely amounts to "contaminating" neighbouring cells, proving the class of convergence was a difficult task. It is only recently that a proof has been proposed by Gerin, who succeeded in applying subtle combinatorial arguments to obtain upper and lower bounds on the time of convergence [Ger17].

The minority rule received a special attention. Indeed, when updated asynchronously, it has the ability to create patterns which can take the form of chequerboard or stripes. The behaviour of this rule with an asynchronous updating was analysed in the case of von Neumann and Moore neighbourhood (the cell and its eight nearest neighbours) [RST09, RST10]. Regnault et al. noticed that the convergence to the fixed point was not uniform: the process can be separated in two phases: first the "energy" decreases rapidly, and then the system stays in a low-energy state where it will progressively approach the fixed point by moving the unstable patterns thanks to the random fluctuations. It is an open question to know to which extent this type of behaviour can be found in other contexts, e.g., lattice-gas cellular automata [BFC13b].

The convergence properties can thus be determined quite precisely, but only for a family of simple binary cellular automata rules. It is an open problem to find such analytical tools. As far as the α-asynchronous updating is concerned, the results are even more restricted. As we will see in the following, this not is so surprising because the behaviour of some rules sometimes require the introduction of tools from advanced statistical physics.

4 Phase transitions induced by α-asynchronous updating

4.1 The Game of Life

We propose to come back to the phenomenon observed on Fig. 1 (see p. 3). Blok and Bergersen were the first authors to give a precise explanation of the change of behaviour in the Game of Life, the phenomenon that was described in the introductory part of this article. They identified the existence of a *second-order phase transition*[4] which separates two qualitatively different behaviours: a high-density steady state with vertical and horizontal stripes and low-density steady state with avalanches [BB99]. They measured the critical value of the synchrony rate at $\alpha_c \approx 0.906$ and showed that near the critical point, the stationary density d_∞ obeyed a power law of the form $d_\infty \sim (\alpha - \alpha_c)^\beta$. It is well known in the field of statistical physics that the values taken by the power laws are not arbitrary and that various systems of unrelated fields may display the same critical exponents ([(See F. Bagnoli's article in this encyclopedia)]. The class of systems which share the same values of exponents is called a *universality class* and in the case of the Game of Life, Blok and Bergersen found that its phase transition was likely to belong to the universality class of *directed percolation* (also called *oriented percolation* or Reggeon field theory).

These measures were later confirmed by a set of more precise experiments [Fat10] and the critical value of the synchrony rate was measured at $\alpha_c \approx 0.911$. Moreover, for the Game of Life, the critical phenomenon was shown to be robust to the introduction of a small degree of irregularity in the grid. This phase transitions was also observed for other Life-like rules [Fat10].

4.2 Elementary cellular automata

In the first experiment where the whole set of ECAs was examined with an α-asynchronous updating [FM05], some rules were observed to display an abrupt variation of the density for a given value of the synchrony rate α. This phenomenon was later studied in detail and this critical phenomenon was identified for ten (non-equivalent) rules. As for the Game of Life, we are here in presence of second-order phase transitions which belongs to the directed percolation universality class [Fat09]. The values of the measured critical synchrony rates are reported on Tab. 3.

It is a puzzling question to know why these ten rules are specifically producing such critical phenomena. Some insights to this question were given in a study of the local-structure approximations of the rules, that is, a generali-

[4] Informally, in statistical physics, phase transitions are defined by the existence of a discontinuity in the values taken by a macroscopic parameter, called the order parameter, when system is submitted to a continuous variation of a control parameter. First-order transitions are those for which the discontinuity appears directly on the order parameter while second-order phase transitions (or *continuous* phase transitions) are those where the derivative of the order parameter is infinite.

Table 3: Critical synchrony rates for the ECA with a phase transition.

ECA	6	18	26	38	50	58	106	134	146	178
α_c	0.283	0.714	0.475	0.041	0.628	0.340	0.815	0.082	0.675	0.410

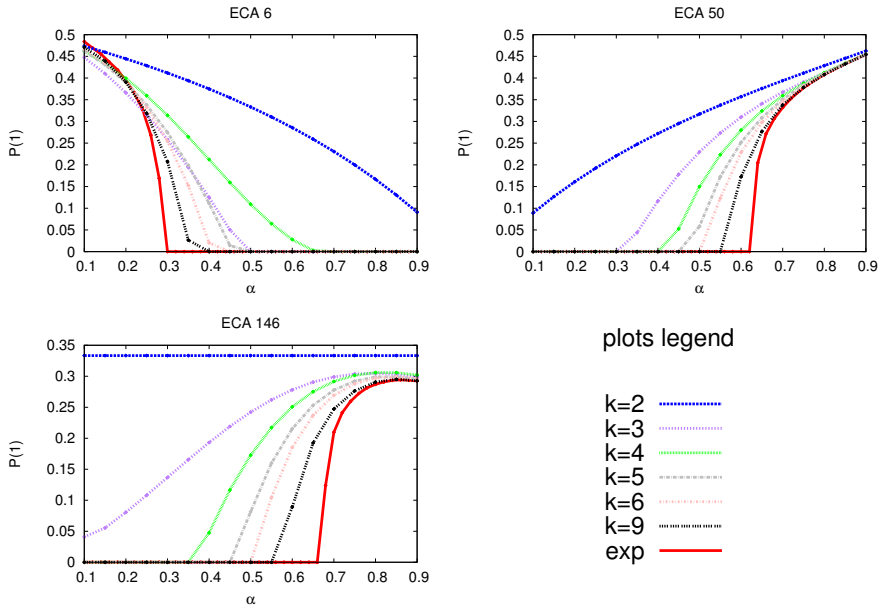


Figure 8: Local-structure approximations obtained for various approximation levels of order k (see Ref. [FF15] for details). For the sake of readability of the results, the cases $k = 7$ and $k = 8$ are omitted. The plot in red (label "exp") shows the experimental steady-state density obtained for a ring size of $n = 40\,000$ cells after $10\,000$ time steps.

sation of the mean-field approximation to correlations of higher order [FF15]. This study revealed that it was possible to predict the occurrence of a phase transition, but it was not possible to use it to correctly approximate the value of the critical synchrony rate. Another possible approach would be to analyse the branching-annihilating phenomenon in a specific way, with small-size Markov chains for instance, but this remains an open path of research.

5 Other questions related to the dynamics

In order to broaden our view of asynchronous cellular automata, we now briefly mention some other problems which have been studied with analytical tools.

Table 4: Wolfram codes and transition codes of the 16 recurrent rules (from Ref. [FSD17]). The two seprate rules are recurrent for $n \neq 3$.

35:ABDEFGH	38:BDFGH	43:ABDEGH	46:BDGH
51:ABCDEFGH	54:BCDFGH	57:ACDEGH	60:CDGH
62:BCDGH	105:ADEH	108:DH	134:BFG
142:BG	150:BCFG	156:CG	204:I
33:ADEFGH	41:ADEGH		

5.1 Reversibility

The question of reversibility amounts to know if it is always possible to "go back in time" and to know if any configuration has a unique predecessor. This question is undecidable in general but there are some sets of rules for which one can tell whether a rule is reversible or not (see Ref. [Mor08] for a survey on this question). In the context of random asynchronous updating, the question cannot be transposed in a direct way because the evolution of the system is not one-to-one (otherwise we would have a deterministic system).

To date, two different approaches have been considered for *finite* systems. Wacker and Worsch proposed to examine the transition graph of the Markov chain of the asynchronous system [WW13]. A rule is said to be phase-space invertible if there exists another rule – possibly itself – whose transition graph is the "inverse" of the graph of the original rule. By "inverse" it is meant that the directions of the transitions are reversed. In other words, the probabilities to go from x to y are identical if one permutes x and y. Interestingly, the authors show that the property of being phase-space invertible is decidable for one-dimensional fully asynchronous cellular automata.

Another approach has been proposed by Sethi et al.: to interpret the reversibility of a system as the possibility to always back to the initial condition [SFD14]. The problem then amounts to deciding the recurrence property of the Markov chain. This allows the authors to propose a partition of the Elementary Cellular Automata according to their recurrence properties and to show that among the 88 non-equivalent rules, there are 16 rules which are recurrent for any ring size greater than two and 2 rules which are recurrent for ring sizes greater than three [FSD17]. These rules are listed in Tab. 4. For the recurrent rules, the structure of the transition graph was analysed as well as the number of connected components of this graph, that is, the number of communication classes of the rules. It was found that the number of classes of communication varies greatly from one rule to another: some rules have an exponential number, while other have a constant number ; the most interesting examples were obtained for the rules with an "intermediary" behaviour. For example for rule 105:ADEH, the number of communication classes is 2 for an odd ring size n and is equal to $n/2+3$ when n is divisible by 4 and to $n/2$ when n is even and not a multiple of 4. It is an open question to generalise these results to other types of rules or to other types of updating schemes.

These results are encouraging and it is rather pleasant to note that contrarily to the problem of convergence seen above, deciding the recurrence properties of an ECA can be achieved. It is thus interesting to see to which extent these results apply to a broader class of systems, including infinite-size systems.

5.2 Coalescence

In the experimental study of the α-asynchronous ECA [FM05], a strange phenomenon was noticed for ECA 46, almost by chance: though this rule does not converge to a fixed point and remains in a chaotic-like steady state, its evolution does not seem to depend on the initial condition. All seems to happen as if the evolution of the rule was only dictated by the sequence of updates that is applied. This phenomenon, named *coalescence*, can be observed on Fig. 9: if we start from two different initial conditions of the same size and apply the same updates on the two systems, they quickly synchronise and adopt the same evolution. This is a particular kind of synchronisation where no desynchronisation is possible: after the coalescence has occurred, the two trajectories remain identical as the local rules are deterministic. The question is to know under which conditions coalescence happens and how long does it take in average for two different initial conditions to "merge" their trajectories.

Rouquier and Morvan have studied experimentally this phenomenon for the 88 ECA with α-asynchronous updating [RM09]. They discovered an unexpected richness of behaviour: some rules coalesce rapidly, other slowly, some never coalesce, some even display phase transitions, etc. Insights have been given by Francès de Mas on this question and a classification of the convergence time has been given from both the observation of space-time diagrams and an analysis of the behaviour [FdM17]. It is still an open question to provide a complete mathematical analysis of these systems and to issue a proof that coalescence can indeed happen in a linear time with respect to the ring size.

5.3 Other problems

There are many other problems which have led to various interesting experimental or theoretical works. For instance, Gacs [Gác01], and then MacAuley and Mortveit [MMM08, MM10, MM13], have provided a deep analysis on the independence of the trajectories of an asynchronous with regard to the updating sequence. Chassaing and Gerin analysed the scaling relationships that would lead to an infinite-size continuous framework [CG07]. This framework is also analysed in detail by Dennunzio et al., who examined how the theory of measure can be applied to one-dimensional systems defined on an infinite line [DFMM13, DFM$^+$17].

As an example of a possible application of the use of these dynamical systems, we mention the work of Das et al., who proposed to use such models for pattern classification [SRD16] and the work of Takada et al., who designed asynchronous self-reproducing loops [TIPM07a]. These are only some entry point to

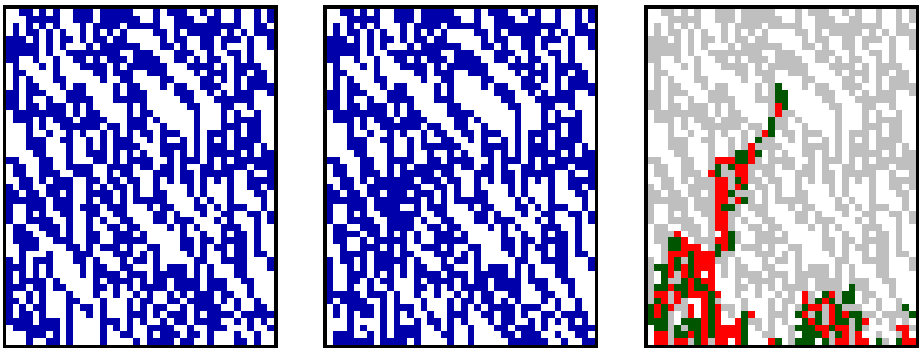


Figure 9: Rapid coalescence phenomenon for ECA 46 with fully asynchronous updating. The same updates are applied on two systems with two different random initial conditions (left) and (middle). The right diagram shows the agreement and disagreement of the two systems. Cells in white and light grey respectively show agreement on state `0` or `1`, while red and green show disagreement (the order is not important).

the literature on this topic and we refer again to our survey paper for a wider scope [Fat14a].

6 Openings

We have seen that the randomness involved in the asynchronous updating create an amazing source of new questions on cellular automata. After more than two decades of continued efforts, this topic shows signs of maturity and, although it remains in large part a *terra incognita*, there are some insights on how asynchronous cellular automata can be studied with a theoretical point of view. A set of analytical tools are now available and when the analysis fails to answer all the question, one can carry out numerical simulations. Readers should now be convinced that asynchronous cellular automata are by no means some "exotic" mathematical objects but constitute a thriving field of research. The elements we presented here are only a small part of this field and should be completed by a more extensive bibliographical work. Before closing this text, we want to present a few questions that are currently investigated.

Defining asynchrony

As mentioned in the introduction, asynchrony is a concept that can be defined with a great variety of forms. For example, the notion of α-asynchronous updating scheme needs to be generalised to go beyond the simple homogeneous finite case. This has led to propose to use some measure-theoretic tools to define *m-asynchronous* cellular automata to include the cases of non-homogeneous probabilities of updating, infinitesimal ones, etc. [DFM12, DFMM13].

To complete this point, let us underline that Bouré et al. have proposed to examine the case where the randomness occurs not on the moments of updating, but on the possibility to miss the information from one or several neighbours [BFC12]. Interestingly, the study of these new updating schemes, named β- and γ-asynchronous updating schemes, show that their behaviour partially overlaps with α-asynchronous systems but also reveals some novel and unexpected behaviours (e.g., other rules show a phase transitions).

Asynchronous models

The theoretical results obtained so far do not tell us what is a *good* model of asynchrony *in general*. Since cellular automata are defined with a *discrete* of time and space, it is not straightforward to decide *a priori* to use a synchronous updating, or a fully asynchronous one, or a partially synchronous one. In fact, the most reasonable position would be to test various updating schemes on a rule and to examine if it is robust or sensitive to these modifications. Although this *critical attitude* has been quite rare so far, a good example of such a study has been provided by Grilo and Correia, who made a systematic study of the effects of the updating in the spatially-extended evolutionary games. This question rose after the criticisms made by Huberman and Glance [HG93] to the model proposed by Nowak and May [NM92]. We think that exploring more systematically these issues on real-world models could help us understand to which extent the simplifications operated in a model are justified or are a potential source of artifacts (see Ref. [Fat14a] for other examples).

Experimental approaches and theoretical questions

The questions of how to measure the behaviour of asynchronous systems is of course primordial. Among the various approaches, let us mention that Silva and Correia have shown the importance of taking into account the time-rescaling effects when using experiments [SC13]. Louis has underlined that the efficiency of a simulation may greatly vary depending on the different regimes that may occur in a simulation [Lou15]. Recently, Bołt et al. have raised the problem of identification: if one is given a space-time diagrams with missing parts, how can one find the rule which generated this piece of information? [BWBB16]. [(See also A. Adamatzky's article in this encyclopedia for the general problem of identification.)].

New models of computation

As mentioned earlier, it is no surprise if the computing abilities of *general* asynchronous cellular automata are the same as those of their deterministic counterparts. However, as shown by various authors, the question becomes much more delicate if one asks how to simulate an asynchronous system by another asynchronous system or if one wants to design asynchronous systems which hold good computing abilities and use a limited number of states [TIPM07b, Wor13].

On the technological side, let us mention the work of Silva et al. on modelling the interactions between (static) robots which need to synchronise [SCC15]. Lee, Peper and their collaborators, who aim at developing asynchronous circuits which are designed with simple local rules [PLAM03]. Such *Brownian cellular automata* [LP08] exploit the inherent fluctuations of the particles to perform asynchronous computations [PLI10, LPLG16]. They represent a potential source of major technical innovations, in particular with the possibility of implementing such circuits with DNA reaction-diffusion systems [YIP+17] or Single Electron Tunnelling techniques [LPC+16].

References

[BB99] Hendrik J. Blok and Birger Bergersen. Synchronous versus asynchronous updating in the "game of life". *Physical Review E*, 59:3876–9, 1999.

[BF12] Selma Belgacem and Nazim Fatès. Robustness of multi-agent models: the example of collaboration between turmites with synchronous and asynchronous updating. *Complex Systems*, 21(3):165–182, 2012.

[BFC12] Olivier Bouré, Nazim Fatès, and Vincent Chevrier. Probing robustness of cellular automata through variations of asynchronous updating. *Natural Computing*, 11:553–564, 2012.

[BFC13a] Olivier Bouré, Nazim Fatès, and Vincent Chevrier. First steps on asynchronous lattice-gas models with an application to a swarming rule. *Natural Computing*, 12(4):551–560, 2013.

[BFC13b] Olivier Bouré, Nazim Fatès, and Vincent Chevrier. A robustness approach to study metastable behaviours in a lattice-gas model of swarming. In Jarkko Kari, Martin Kutrib, and Andreas Malcher, editors, *Proceedings of Automata'13*, volume 8155 of *Lecture Notes in Computer Science*, pages 84–97. Springer, 2013.

[BI84] Buvel, R.L. and Ingerson, Thomas E. Structure in asynchronous cellular automata. *Physica D*, 1:59–68, 1984.

[BWBB16] Witold Bołt, Barbara Wolnik, Jan M. Baetens, and Bernard De Baets. On the identification of α-asynchronous cellular automata in the case of partial observations with spatially separated gaps. In Guy De Tré, Przemyslaw Grzegorzewski, Janusz Kacprzyk, Jan W. Owsinski, Wojciech Penczek, and Slawomir Zadrozny, editors, *Challenging Problems and Solutions in Intelligent Systems*, pages 23–36. Springer, 2016.

[CF10] Vincent Chevrier and Nazim Fatès. How important are updating schemes in multi-agent systems? An illustration on a multi-turmite

model. In *Proceedings of AAMAS '10*, pages 533–540, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[CG07] Philippe Chassaing and Lucas Gerin. Asynchronous cellular automata and brownian motion. In *DMTCS Proceedings of AofA'07*, volume AH, pages 385–402, 2007.

[CGN05] David Cornforth, David G. Green, and David Newth. Ordered asynchronous processes in multi-agent systems. *Physica D*, 204(1–2):70 – 82, 2005.

[DFM12] Alberto Dennunzio, Enrico Formenti, and Luca Manzoni. Computing issues of asynchronous CA. *Fundamenta Informaticae*, 120(2):165–180, 2012.

[DFM+17] Alberto Dennunzio, Enrico Formenti, Luca Manzoni, Giancarlo Mauri, and Antonio E. Porreca. Computational complexity of finite asynchronous cellular automata. *Theoretical Computer Science*, 664:131–143, 2017.

[DFMM13] Alberto Dennunzio, Enrico Formenti, Luca Manzoni, and Giancarlo Mauri. m-asynchronous cellular automata: from fairness to quasi-fairness. *Natural Computing*, 12(4):561–572, 2013.

[dO14] Pedro P. B. de Oliveira. On density determination with cellular automata: Results, constructions and directions. *Journal of Cellular Automata*, 9(5-6):357–385, 2014.

[Fat09] Nazim Fatès. Asynchronism induces second order phase transitions in elementary cellular automata. *Journal of Cellular Automata*, 4(1):21–38, 2009.

[Fat10] Nazim Fatès. Does *life* resist asynchrony? In Andrew Adamatzky, editor, *Game of Life Cellular Automata*, pages 257–274. Springer London, 2010.

[Fat13a] Nazim Fatès. A note on the classification of the most simple asynchronous cellular automata. In Jarkko Kari, Martin Kutrib, and Andreas Malcher, editors, *Proceedings of Automata'13*, volume 8155 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 2013.

[Fat13b] Nazim Fatès. Stochastic cellular automata solutions to the density classification problem - when randomness helps computing. *Theory of Computing Systems*, 53(2):223–242, 2013.

[Fat14a] Nazim Fatès. A guided tour of asynchronous cellular automata. *Journal of Cellular Automata*, 9(5-6):387–416, 2014.

[Fat14b] Nazim Fatès. Quick convergence to a fixed point: A note on asynchronous elementary cellular automata. In Jarosław Was, Georgios Ch. Sirakoulis, and Stefania Bandini, editors, *Proceedings of ACRI'14*, volume 8751 of *Lecture Notes in Computer Science*, pages 586–595. Springer, 2014.

[FdM17] Jordina Francès de Mas. Coalescence in fully asynchronous elementary cellular automata. Technical report, HAL preprint hal-01627454, 2017.

[FF15] Henryk Fukś and Nazim Fatès. Local structure approximation as a predictor of second-order phase transitions in asynchronous cellular automata. *Natural Computing*, 14(4):507–522, 2015.

[FG09] Nazim Fatès and Lucas Gerin. Examples of fast and slow convergence of 2D asynchronous cellular systems. *Journal of Cellular Automata*, 4(4):323–337, 2009.

[FM05] Nazim Fatès and Michel Morvan. An experimental study of robustness to asynchronism for elementary cellular automata. *Complex Systems*, 16:1–27, 2005.

[FMST06] Nazim Fatès, Michel Morvan, Nicolas Schabanel, and Eric Thierry. Fully asynchronous behavior of double-quiescent elementary cellular automata. *Theoretical Computer Science*, 362:1–16, 2006.

[FRST06] Nazim Fatès, Damien Regnault, Nicolas Schabanel, and Eric Thierry. Asynchronous behavior of double-quiescent elementary cellular automata. In José R. Correa, Alejandro Hevia, and Marcos A. Kiwi, editors, *Proceedings of LATIN 2006*, volume 3887 of *Lecture Notes in Computer Science*, pages 455–466. Springer, 2006.

[FSD17] Nazim Fatès, Biswanath Sethi, and Sukanta Das. On the reversibility of ecas with fully asynchronous updating: the recurrence point of view. To appear in a monography edited by Andrew Adamatzky – Preprint available on the HAL server, id: `hal-01571847`, 2017.

[Fuk02] Henryk Fukś. Nondeterministic density classification with diffusive probabilistic cellular automata. *Physical Review E*, 66(6):066106, 2002.

[Gác01] Peter Gács. Deterministic computations whose history is independent of the order of asynchronous updating. Technical report - arXiv:cs/0101026, 2001.

[Ger17] Lucas Gerin. Epidemic automaton and the eden model: various aspects of robustness. Text to appear in a monography on probabilistic cellular automata (Springer), 2017.

[HG93] Bernardo A. Huberman and Natalie Glance. Evolutionary games and computer simulations. *Proceedings of the National Academy of Sciences, USA*, 90:7716–7718, 1993.

[KT15] Jarkko Kari and Siamak Taati. Statistical mechanics of surjective cellular automata. *Journal of Statistical Physics*, 160(5):1198–1243, 2015.

[LAPM04] Jia Lee, Susumu Adachi, Ferdinand Peper, and Kenichi Morita. Asynchronous game of life. *Physica D*, 194(3–4):369–384, 2004.

[Lou15] Pierre-Yves Louis. Supercritical probabilistic cellular automata: how effective is the synchronous updating? *Natural Computing*, 14(4):523–534, 2015.

[LP08] Jia Lee and Ferdinand Peper. On brownian cellular automata. In Andrew Adamatzky, Ramón Alonso-Sanz, Anna T. Lawniczak, Genaro Juárez Martínez, Kenichi Morita, and Thomas Worsch, editors, *Proceedings of Automata 2008*, pages 278–291. Luniver Press, Frome, UK, 2008.

[LPC+16] Jia Lee, Ferdinand Peper, Sorin Dan Cotofana, Makoto Naruse, Motoichi Ohtsu, Tadashi Kawazoe, Yasuo Takahashi, Tetsuya Shimokawa, Laszlo B. Kish, and Tohru Kubota. Brownian circuits: Designs. *International Journal of Unconventional Computing*, 12(5-6):341–362, 2016.

[LPLG16] Jia Lee, Ferdinand Peper, Kenji Leibnitz, and Ping Gu. Characterization of random fluctuation-based computation in cellular automata. *Information Sciences*, 352-353:150–166, 2016.

[MM10] Matthew Macauley and Henning S. Mortveit. Coxeter groups and asynchronous cellular automata. In Stefania Bandini, Sara Manzoni, Hiroshi Umeo, and Giuseppe Vizzari, editors, *Proceedings of ACRI'10*, volume 6350 of *Lecture Notes in Computer Science*, pages 409–418. Springer, 2010.

[MM13] M. Macauley and H.S. Mortveit. An atlas of limit set dynamics for asynchronous elementary cellular automata. *Theoretical Computer Science*, 504(0):26 – 37, 2013. Discrete Mathematical Structures: From Dynamics to Complexity.

[MM14] Jean Mairesse and Irène Marcovici. Around probabilistic cellular automata. *Theoretical Computer Science*, 559(0):42–72, 2014. Non-uniform Cellular Automata.

[MMM08] Matthew Macauley, Jon McCammond, and Henning S. Mortveit. Order independence in asynchronous cellular automata. *Journal of Cellular Automata*, 3(1):37–56, 2008.

[Moo62] Edward F. Moore. Machine models of self-reproduction. *Proceedings of Symposia in Applied Mathematics*, 14:17–33, 1962. (Reprinted in Essays on Cellular Automata, A.W. Burks (ed.), University of Illinois Press, 1970).

[Mor08] Kenichi Morita. Reversible computing and cellular automata — a survey. *Theoretical Computer Science*, 395(1):101 – 131, 2008.

[Nak74] Katsuhiko Nakamura. Asynchronous cellular automata and their computational ability. *Systems, Computers, Controls*, 5(5):58–66, 1974.

[Nak81] Katsuhiko Nakamura. Synchronous to asynchronous transformation of polyautomata. *Journal of Computer and System Sciences*, 23(1):22 – 37, 1981.

[NM92] Martin A. Nowak and Robert M. May. Evolutionary games and spatial chaos. *Nature*, 359:826–829, 1992.

[PLAM03] Ferdinand Peper, Jia Lee, Susumu Adachi, and Shinro Mashiko. Laying out circuits on asynchronous cellular arrays: a step towards feasible nanocomputers? *Nanotechnology*, 14(4):469, 2003.

[PLI10] Ferdinand Peper, Jia Lee, and Teijiro Isokawa. Brownian cellular automata. *Journal of Cellular Automata*, 5(3):185–206, 2010.

[Reg13] Damien Regnault. Proof of a phase transition in probabilistic cellular automata. In Marie-Pierre Béal and Olivier Carton, editors, *Proceedings of Developments in Language Theory*, volume 7907 of *Lecture Notes in Computer Science*, pages 433–444. Springer, 2013.

[RL17] Alex Dias Ramos and André Leite. Convergence time and phase transition in a non-monotonic family of probabilistic cellular automata. *Journal of Statistical Physics*, 168(3):573–594, 2017.

[RM09] Jean-Baptiste Rouquier and Michel Morvan. Coalescing cellular automata: Synchronization by common random source for asynchronous updating. *Journal of Cellular Automata*, 4(1):55–78, 2009.

[RST09] Damien Regnault, Nicolas Schabanel, and Eric Thierry. Progresses in the analysis of stochastic 2D cellular automata: A study of asynchronous 2D minority. *Theoretical Computer Science*, 410(47-49):4844–4855, 2009.

[RST10] Damien Regnault, Nicolas Schabanel, and Eric Thierry. On the analysis of "simple" 2d stochastic cellular automata. *Discrete Mathematics & Theoretical Computer Science*, 12(2):263–294, 2010.

[SC13] Fernando Silva and Luís Correia. An experimental study of noise and asynchrony in elementary cellular automata with sampling compensation. *Natural Computing*, 12(4):573–588, 2013.

[SCC15] Fernando Silva, Luís Correia, and AndersLyhne Christensen. Modelling synchronisation in multirobot systems with cellular automata: Analysis of update methods and topology perturbations. In Georgios Ch. Sirakoulis and Andrew Adamatzky, editors, *Robots and Lattice Automata*, volume 13 of *Emergence, Complexity and Computation*, pages 267–293. Springer International Publishing, 2015.

[SdR99] Birgitt Schönfisch and André de Roos. Synchronous and asynchronous updating in cellular automata. *BioSystems*, 51:123–143, 1999.

[SFD14] Biswanath Sethi, Nazim Fatès, and Sukanta Das. Reversibility of elementary cellular automata under fully asynchronous update. In T.V. Gopal, Manindra Agrawal, Angsheng Li, and Barry Cooper, editors, *Proceedings of TAMC'14*, volume 8402 of *Lecture Notes in Computer Science*, pages 39–49. Springer, 2014.

[SRD16] Biswanath Sethi, Souvik Roy, and Sukanta Das. Asynchronous cellular automata and pattern classification. *Complexity*, 21:370–386, 2016.

[TIPM07a] Yousuke Takada, Teijiro Isokawa, Ferdinand Peper, and Nobuyuki Matsui. Asynchronous self-reproducing loops with arbitration capability. *Physica D: Nonlinear Phenomena*, 227(1):26 – 35, 2007.

[TIPM07b] Yousuke Takada, Teijiro Isokawa, Ferdinand Peper, and Nobuyuki Matsui. Asynchronous self-reproducing loops with arbitration capability. *Physica D*, 227(1):26 – 35, 2007.

[Vic84] Gérard Y. Vichniac. Simulating physics with cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):96 – 116, 1984.

[Vie13] Michael Vielhaber. Computation of functions on n bits by asynchronous clocking of cellular automata. *Natural Computing*, 12(3):307–322, 2013.

[Wol85] Stephen Wolfram. Twenty problems in the theory of cellular automata. *Physica Scripta*, 1985(T9):170, 1985.

[Wor13] Thomas Worsch. Towards intrinsically universal asynchronous CA. *Natural Computing*, 12(4):539–550, 2013.

[WW13] Simon Wacker and Thomas Worsch. On completeness and decidability of phase space invertible asynchronous cellular automata. *Fundamenta Informaticae*, 126(2-3):157–181, 2013.

[YIP+17] Tatsuya Yamashita, Teijiro Isokawa, Ferdinand Peper, Ibuki Kawamata, and Masami Hagiya. Turing-completeness of asynchronous non-camouflage cellular automata. In Alberto Dennunzio, Enrico

Formenti, Luca Manzoni, and Antonio E. Porreca, editors, *Processdings of AUTOMATA 2017*, volume 10248 of *Lecture Notes in Computer Science*, pages 187–199. Springer, 2017.