# Remarks on the cellular automaton global synchronisation problem – deterministic vs. stochastic models

Nazim A. Fatès

## To cite this version:

Nazim A. Fatès. Remarks on the cellular automaton global synchronisation problem – deterministic vs. stochastic models. Natural Computing, 2019, 18 (3), pp.429-424. 10.1007/s11047-018-9683-0 . hal-01653631v2

HAL Id: hal-01653631
https://inria.hal.science/hal-01653631v2

Submitted on 9 Apr 2018

# Remarks on the cellular automaton global synchronisation problem

Nazim Fatès
Université de Lorraine, CNRS, Inria, LORIA,
F-54000 Nancy, France
nazim.fates@inria.fr

April 9, 2018

**Abstract**

In the global synchronisation problem, one is asked to find a cellular automaton which has the property that every initial condition evolves into a homogeneous blinking state. We study this simple inverse problem for the case of one-dimensional systems with periodic boundary conditions. Two paradoxical observations are made: (a) despite the apparent simplicity of finding rules with good statistical results, there exist no perfect deterministic solutions to this problem, (b) if we allow the use of randomness in the local rule, constructing "perfect" stochastic solutions is easy. For the stochastic case, we give some rules for which the mean time of synchronisation varies quadratically with the number of cells and ask if this result can be improved. To explore more deeply the deterministic rules, we code our problem as a SAT problem and USE SAT solvers to find rules that synchronise a large set of initial conditions (in appendix).

## 1 Introduction

Simple inverse problems play an important role in the field of cellular automata: they stimulate the birth of new ideas and invite researchers to exchange various points of view on the same questions. Among the recent achievements related to such problems, we may mention the construction of an exact solution to the parity problem [BdOF13], the construction of exact or approached solutions on the density classification problem [Fat13, BFMM13, BnMdEOR13, dO14] or the re-interpretation of the solution to the Firing squad problem with fields [MY14].

A common feature of the simple cellular automata inverse problems is that they require reaching a state of global consensus: there should be a point where all the cells, or a large fraction of cells, agree on a state; this state is then interpreted as the output of the cellular algorithm. The difficulty for solving these problems is often related to the propagation of information from a local to a global scale: in a decentralised framework such as cellular automata, the main question is how the cells may "agree" on a common state when they have only a local view of the system.

In this text, we study the *global synchronisation problem*: in its original form, the problem consists in finding a rule such that the system reaches a "blinking state" from any initial condition. This blinking state is made of two homogeneous configurations (all-0 and then all-1) which alternate. This problem is also generalised for more than two states, and the blinking condition is translated to a cycling condition.

The first scientific account of a phenomenon of mutual synchronisation by loosely coupled components is due to Huyghens (1629-1695). The motivation of Huyghens was to build clocks that would be sufficiently precise to help sailors perform a measure of the longitude. He was looking for a means to obtain a stable beating pace for one or more clocks, even if the clocks were put on a ship. In the important book dedicated to synchronisation [PRK03], Pikovsky et al. relate that, in a letter to his father dated February 1665, Huyghens noticed that two pendulum clocks that were hanging on a wall had a tendency to swing in pace. He called this the "sympathy of two clocks" (*sympathie des horloges*) and proposed a first simple explanation on how such a surprising phenomenon happens. He could also observe this phenomenon for two pendulums that were hung on a wooden support and he proposed that such a mechanism would be developed to ensure the stability of the beat of the clocks that were taken aboard the ships. Unfortunately, this idea could not be put in practice because it was not possible to stabilize the clocks for the duration of a whole sea crossing (several weeks or months).

Since then, synchronisation phenomena have been observed in a wide range of sciences. The term synchronisation has been used with different meanings. In our context, we are mainly interested in *self-synchronisation*, that is, the development of a common form of oscillation between weakly coupled parts *without the application of an external stimulus*. It is not always easy to distinguish between self- and non self-synchronisation but the examples that are often given are the flashing of some species of fireflies, the walk movements of pedestrians, the common clapping of a public after a concert, etc. (see Ref. [PRK03] for an overview). In the realm of cellular biology, to cite only one example among many, let us also mention that it has been discovered that the two flagella that some cells use to "swim" in a liquid are not directly coupled but they simply adjust one to another according to the movements of the cell [Fri16].

In the field of cellular automata, synchronisation has also been studied, but with a different meaning for different authors. In general, the word has been used in the context of coupling: two configurations evolve according to a local rule but the state of some cells are set equal with some external process, which is usually random (see e.g. [BR99, RM09] and references therein).

As mentioned earlier, the problem we address here is different. The system evolves with no other influence than its local evolution rule and we want to observe a phenomenon of self-organisation that would always lead to a collective form of blinking. In some sense, when the blinking occurs, all the cells are in phase and one says that they are *synchronised*. Of course, one may also argue that there is a global clock that already gives the pace of the transitions. This statement is perfectly accurate but it does not explain how the *collective* phenomenon of synchronisation occurs. As we will see, the major difficulty relies in the use of a local rule that is homogeneous in space and time: some kind of symmetry breaking is needed if one wants to form a consensus. We can also note that the problem considerably differs from the Firing Squad problem, where we want all the cells to enter in a particular state simultaneously, but with a particular initial condition (or a small set of initial conditions), and by strictly forbidding to reach this "firing" state before the synchronisation step.

The global synchronisation problem was formulated by Das et al. in 1994 [DCMH95] and has received a quite limited attention since then. This lack of interest may stem from the fact that at first sight, the problem seems easy to solve, at least much easier than other inverse problems such as the density classification problem [dO14]. In fact, good or "almost perfect" solutions were presented in the very paper where the problem was formulated [DCMH95]. The authors used genetic algorithms to exhibit cellular automata rules with a performance of "100%" for ring sizes going up to 999. Their interest was not so much on the performance, but on using genetic algorithms to find rules which show an "emerging" phenomenon, here to attain a consensus by removing the "defects" that separate the non-synchronised regions of the system.

Our purpose now is to go one step forward and to examine if perfect solutions do exist and how they can be obtained. The requirement is that *all* initial conditions lead to the homogeneous cycle and not only a fraction of configurations. After giving some formal definitions (Sec. 2), we analyse what can be achieved with different settings of the problem, in particular we consider the case where the use of auxiliary states is allowed (Sec. 3). We also present stochastic solutions which are "perfect" solutions in the sense that they converge almost surely to a synchronised state (Sec. 4). We then briefly discuss these observations and conclude with a few questions (Sec. 5). In the appendix, we present how to explore the space of deterministic rules by using SAT solvers (Sec. 6)[1].

# 2 Fundamentals

## 2.1 CA definitions

The basic components of our systems are the *cells* ; we consider here only the *one-dimensional finite case* but most constructions also apply in higher dimensions. The cells are spatially arranged on a line with periodic boundary conditions ; their number is denoted by $n$ and, since their spatial arrangement is equivalent to a ring, we denote the set of cells by $\mathcal{L} = \mathbb{Z}/n\mathbb{Z}$, the classes of equivalence modulo $n$.

Each cell can hold a state taken in a set $Q_k$ of cardinal $k$. Without loss of generality, we can consider that $Q_k = \mathbb{Z}/k\mathbb{Z}$, that is, we have $k$ states which are

[1] This article is an extended version of a first text presented at AUTOMATA 2015 [Fat15].

represented by the integers 0 to $k-1$, and the set of states is endowed with the addition modulo $k$.

A *configuration* $x = (x_i)_{i \in \mathcal{L}}$ represents the state of the system at a given time. The set of configurations is denoted by $\mathcal{E}_n = {Q_k}^{\mathcal{L}}$. The interactions between the cells are local, that is, each cell can only "see" a finite subset of the cells. This relationship is given by the *neighbourhood*, denoted by $\mathcal{N}$ and defined as a sequence of translations in $\mathcal{L}$ from the cell to its neighbours. We write: $\mathcal{N} = \{\nu_1, \ldots, \nu_s\} \in \mathcal{L}^s$, where $s \in \mathbb{N}$ is the *size* of the neighbourhood.

The evolution of a cell follows a function $f : {Q_k}^s \to Q_k$, called the *local rule*.

For a given ring size $n$, the local rule allows us to define the *global transition function* $F : \mathcal{E}_n \to \mathcal{E}_n$, that is, the function that maps a configuration $x^t$ to a configuration $x^{t+1} = F(x^t)$ such that $x^0 = x$ and:

$$\forall i \in \mathcal{L},\ x_i^{t+1} = f\big((x_{i+\nu_j}^t)_{j \in \{1,\ldots,s\}}\big). \tag{1}$$

Note that to be perfectly rigorous, we should denote $F$ with indices showing that it depends on $f$ and $\mathcal{N}$. We however drop these elements for the sake of simplicity since they are in general clear from the context. Note that we will also use the notation $x^t = F^t(x)$ with the usual sense of the composition.

## 2.2 Formulation of the problem

For a state $q \in Q_k$, we denote by $\mathbf{q} = q^{\mathcal{L}}$ the *homogeneous* configuration with state $q$ and $\mathcal{H} = \{\mathbf{q}, q \in Q_k\}$ the set of the $k$ homogeneous configurations.

We say that a global transition function $F$ *synchronises* a configuration $x \in \mathcal{E}_n$ if the system with the initial condition $x$ reaches a homogeneous configuration and then cycles through $\mathcal{H}$ in order ; formally:

$$\exists q \in Q_k, t \in \mathbb{N}; F^t(x) = \mathbf{q} \text{ and } \forall \tau \in \mathbb{N},\ F^{t+\tau} = (\mathbf{q} + \boldsymbol{\tau}) = (q+\tau)^{\mathcal{L}}.$$

We say that $F$ synchronises the size $n$ if it synchronises all the configurations of $\mathcal{E}_n$.

Note that the demand for having a cycle of length $k$ is a strong constraint: the rule cannot use any additional state to achieve the synchronisation. Indeed, as we will see below, relaxing this constraint makes the problem rather easy to solve.

In contrast, also note that the request for cycling through $\mathbb{Z}/k\mathbb{Z}$ in order does not bring any loss of generality. Indeed, if we have a global function $F$ which synchronises a given configuration to a cycle of homogeneous configurations $\mathbf{q_1}$, ..., $\mathbf{q_k}$, then, every other configuration which is synchronised necessarily follows the same order. Consequently, if we have a rule $F$ which synchronises all the configurations to a length-$k$ cycle of homogeneous states in an order different from the one of $Q_k$, then we can construct a rule $F'$ which cycles in the right order in $\mathbb{Z}/k\mathbb{Z}$ by a simply reordering the states $q_1, \ldots, q_k$ into $0, \ldots, k-1$.

We can now formulate the global synchronisation problem:

*Find a local rule $f$ such that for every ring size $n$, the global function $F$ synchronises $\mathcal{E}_n$.*

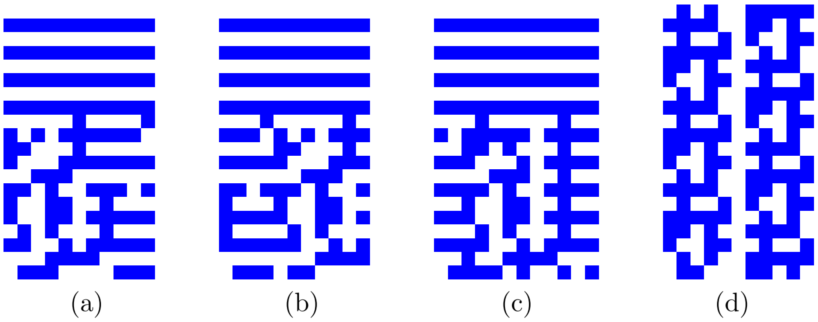

Figure 1: Four space-time diagrams for rule 1078270911 with $k = 5$ (see page 28). Time goes from bottom to top, white and blue squares represent cells in state `0` and `1`, respectively. (a) and (b): synchronised initial conditions with $n = 11$, (c) and (d): initial conditions with $n = 12$, the last one enters a cycle of length 6 and is not synchronised.

### 2.3 Elementary results

Let $f$ be a local rule of size $s$. We now expose the elementary properties that can be given about this rule with regard to the synchronisation problem.

The *reflected* rule $f_{\mathrm{R}}$ is the local rule obtained from $f$ by the exchange of the left and right directions:

$$\forall q_1, \ldots, q_s \in Q_k, f_{\mathrm{R}}(q_1, \ldots, q_s) = f(q_s, \ldots, q_1).$$

For $\delta \in \{1, \ldots, k-1\}$, the $\delta$-conjugated rule $f_\delta$ is the local rule obtained from $f$ by shifting the states by $\delta$ in the set of states $Q_k = \mathbb{Z}/k\mathbb{Z}$,

$$\forall q_1, \ldots, q_s \in Q_k, f_\delta(q_1 + \delta, \ldots, q_s + \delta) = f(q_1, \ldots, q_s) + \delta.$$

For the binary case $k = 2$, this operation is equivalent to the classical conjugation, which corresponds to a global inversion of the `0`'s and the `1`'s.

**Proposition 1** (rule symmetries)**.** *$f$ is a solution to the global synchronisation problem then $f_{\mathrm{R}}$, $f_\delta$ and $f_{\mathrm{R}} \circ f_\delta$ for $\delta \in \{1, \ldots, k-1\}$ are also solutions.*

Clearly, the property of synchronising a given size is preserved by the reflection and conjugation symmetries and by the composition of these operators.

Let $\sigma$ denote the (left) shift operator, that is, the function $\sigma : \mathcal{E}_n \to \mathcal{E}_n$ such that $\forall i \in \mathcal{L}, \sigma(x)_i = x_{i+1}$. We call the *rotations* of $x$ the set of configurations that are obtained by applying a positive number of shifts on $x$; this set is denoted by $[x] = \{\sigma^m(x), m \in \mathbb{N}\} = \{\sigma^m(x), m \in \{0, \ldots, n-1\}\}$.

**Proposition 2** (configuration symmetries)**.** *A global rule $F$ synchronises a configuration $x$ if and only if it synchronises all the configurations of $[x]$.*

This simply results from the fact that: (a) $F$ commutes with the shift $\sigma$ and (b) $[\mathbf{q}] = \{\mathbf{q}\}$. Note that the rotation $[\cdot]$ defines an equivalence class: we say that a configuration $y$ is equivalent to $x$ if $y$ is a rotation of $x$. It can be easily verified that this is an equivalence relation.

The next proposition states that the iterates of a configuration cannot be contained in the rotations of this configuration.

**Proposition 3** (images of a configuration)**.** *If $F$ synchronises a configuration $x \notin \mathcal{H}$, then*

$$\forall x' \in [x],\, \forall t \in \mathbb{N}^*,\, F^t(x') \notin [x].$$

*Proof.* By contradiction, let us assume that there exists $\tau \in \mathbb{N}^*$ and $i \in \mathbb{N}$ such that $F^\tau(x) = \sigma^i(x)$. By recurrence, using the commutation of the shift with $F$, we have: $F^{n\tau}(x) = \sigma^{in}(x)$. Since the space is a ring of size $n$, we have $\sigma^{in}(x) = x$, which implies $x = F^{kn}(x)$. The configuration $x$ thus evolves cyclically. The homogeneous configurations are excluded from this cycle – otherwise $x$ would be reachable from these states – and $x$ cannot be synchronised, which contradicts the hypothesis. □ □

An immediate consequence of this proposition is that if $F$ synchronises $x$, then $x$ cannot be a *fixed point* ($F(x) = x$) or a *translating configuration* ($F(x) = \sigma^i x$ with $i \in \mathbb{N}^*$).

We say that $f$ is *colour-blind* [ST14] if for any permutation $\pi$ of $Q_k$, we have $\forall (q_1, \dots, q_s) \in {Q_k}^s,\ f(\pi(q_1), \dots, \pi(q_s)) = \pi(f(q_1, \dots, q_k))$. In words, this means that the rule is unable to discriminate between the different states that compose its transitions.

**Proposition 4** (colour-discrimination)**.** *If a local rule $f$ is a solution to the problem, then $f_\delta \neq f$ for all $\delta \in \{1, \dots, k-1\}$. In particular, $f$ cannot be colour-blind.*

*Proof.* By contradiction, let us assume that $f$ is a solution and that $\exists \delta \in \{1, \dots, k-1\}, f_\delta = f$.

First, let us assume that the rule is invariant with one state shift, that is, we have $f_1 = f$. We consider the configuration $x$ of size $k$ composed by all the states in order: $x = 012\dots k$. Let us assume that $x$ is synchronised on $\mathbf{q}$: there exists $T \in \mathbb{N}$ such that $F^T(x) = \mathbf{q}$. Let $\mathbb{S} : \mathcal{E}_n \to \mathcal{E}_n$ be the state-shift operator, defined with: $\forall x \in \mathcal{E}_n, \forall i \in \mathcal{L}; \mathbb{S}(x)_i = x_i + 1$. Then, if we consider $\mathbb{S}(x)$, we have:

$$F_c^T(\mathbb{S}(x)) = \mathbb{S}(F^T(x)) = \mathbb{S}(\mathbf{q}) = (\mathbf{q} + \mathbf{1}).$$

On the other hand, since $\mathbb{S}(x) = \sigma x$, we have:

$$F_c^T(\mathbb{S}(x)) = F^T(\sigma x) = \sigma F^T(x) = \sigma \mathbf{q} = \mathbf{q},$$

which leads to a contradiction. The generalisation of this argument to the impossibility to have $f_\delta = f$ for any $\delta$ is straightforward: one simply needs to replace $\sigma$ by $\sigma^\delta$.

□ □

# 3 Theoretical results on deterministic CA

## 3.1 A negative result

**Lemma 1** ([Ric17]). *Every cellular automaton possesses either a fixed point or a translating configuration, that is:* $\exists n \in \mathbb{N}^*, x \in \mathcal{E}_n, i \in \mathbb{N}; F(x) = \sigma^i(x)$.

The first proof of this lemma was proposed by Richard [Ric17][2]. Independently of Richard's discovery, an identical argument was also given to us by Redeker, who was obviously not aware of the existence of a proof. Informally, the idea of Richard and Redeker is to reason by contradiction, and to show that if a rule were a solution to the problem, it would necessarily also possess a translating configuration, which is impossible because such a configuration cannot be synchronised.

To construct this translating configuration, a particular string is built. Assume the rule is of size $s$, then $s$ symbols are arbitrarily generated with binary states; we then compute the result of the local rule on these $s$ symbols and add this symbol at the right of the sequence. This operation is iterated by using the $s$ rightmost symbols of the string, until the process enters in a loop. If one observes the repetitive part of this loop, one sees that, by construction, it forms a translating configuration which is different from the all-zero and the all-one configuration.

We now expose formally this idea of Richard and Redeker; the only difference is that we will use an adaptation of de Bruijn diagrams instead of strings.

*Proof.* Let $f$ be a rule with a neighbourhood of size $s$. Consider the *translation-building graph* $G = (V, E)$ defined as follows:

- the set of vertices $V = Q^s$ corresponds to the neighbourhood states,
- the edges of the graph connect each node $v$ with the node obtained by suppressing the leftmost symbol of $v$ and adding $f(v)$ on the right of $v$. In other words, $(v, v') \in E$ if and only if $v'_i = v_{i+1}$ for all $i \in \{1, \dots, s-1\}$ and $v'_s = f(v) = f(v_1, \dots, v_s)$. This edge is given the label $f(v)$.

This directed graph is formed by $k^s$ vertices and each node is linked to exactly one node, which implies that the graph necessarily contains a cycle of length $l \leq k^s$ (the same argument as above can be used to prove this property). Let $c$ be the configuration of size $l$ in ${Q_k}^{\mathbb{Z}/l.\mathbb{Z}}$ obtained by concatenating the labels of this cycle, then, by construction, $F(c) = \sigma^{s-1}(c)$, that is, $c$ is a translating configuration, which implies that $F$ cannot achieve the synchronisation of $c$. □

□

These translation-building graphs are a subset of de Bruijn diagrams; their construction allows one to easily see the cycles that exist in a rule of an arbitrary neighbourhood size. Examples are given in Figure 2 for Elementary

[2] Richard exposed his idea to us during the AUTOMATA 2015 workshop and where a first presentation of this work was made [Fat15].

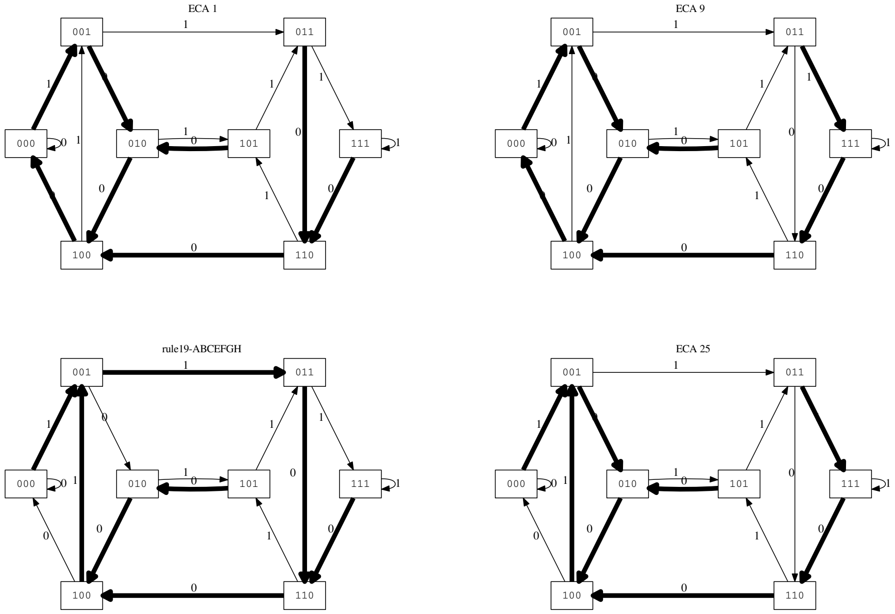

Figure 2: Four examples of translation-building graphs for the Elementary Cellular Automata 1, 9, 19 and 25. Only the "strong" edges belong to the translation-building graph; "thin" edges are shown only to display the de Bruijn diagram that supports this graph. The first three rules have a cycle of length four (resp. `0100`, `0100`, `0110`) and no smaller cycle. ECA 25 has a cycle of length three (`001`).

Cellular Automata (ECA), that is, rules defined with a binary alphabet and with the neighbourhood $\mathcal{N} = \{-1, 0, 1\}$. For the ECA rules denoted by the usual decimal codes 1, 9 and 19, the translating configuration is of size 4 and all the configurations of size 2 and 3 are correctly synchronised (for more details see Sec. 6.2.1).

**Theorem 1.** *There exists no perfect solution to the global synchronisation problem.*

This is a direct consequence of the lemma (see Prop. 3). It is important to note that the theorem above implies that a CA rule of size $s$ cannot synchronise all the ring sizes from 1 to $k^s$, but that it does not imply any limit on the maximum size that can be synchronised. Indeed, it may well be that the translation-building graph contains only a cycle of, say, length 4 but that all the configurations of size 5 are well synchronised. This is for instance the case for ECA 9. It thus an open problem to know if there exists a rule which would synchronise an infinity of ring sizes (e.g., those of prime or odd size).

This lemma and theorem still apply for higher dimensions. Indeed, if we have a $d$-dimensional rule, then by considering the set of configuration which are invariant along $d-1$ dimensions, the effect of the global rules on such configurations is identical to a one-dimensional rule.

## 3.2 Using auxiliary states

We now give some positive results, showing that the use of auxiliary states allows one to solve the synchronisation problem quite simply. By *auxiliary state*, we mean that the state set of the cells is extended with some states that are not in $Q_k$. If we go back to the physical analogy that motivates the synchronisation problem, since the states of $Q_k$ model the discrete "phase" of the oscillating cells, these extra states model a *temporary* external intervention to modify this phase. At the end, these auxiliary states should have disappeared and the system should oscillate within $\mathcal{H}$, the homogeneous states of $Q_k$.

In this part, we will thus consider that the state space is $Q'_k = Q_k \cup Q_{\mathtt{A}}$ and try to make $Q_{\mathtt{A}}$ as small as possible. For a given ring size $n$, the set of configurations is thus ${Q'_k}^{\mathcal{L}}$ but the initial conditions are always taken in $\mathcal{E}_n = {Q_k}^{\mathcal{L}}$ and the cycle formed by homogeneous configurations of $Q_k$.

### 3.2.1 Binary case $k = 2$

The following construction shows that only one auxiliary state is sufficient to "force" locally the configurations to be settled on a given phase. This auxiliary state is made to appear in the border regions, that is, in the cells whose state differs from the state of its neighbours.

We use the state set $Q'_2 = \{\mathtt{0}, \mathtt{1}, \mathtt{A}\}$ and define $f_{\mathrm{A}} : {Q'_2}^3 \to Q'_2$ the following local rule:

$$f_{\mathrm{A}}(x,y,z) = \begin{cases} \mathtt{0} & \text{if } (x,y,z) = (\mathtt{1},\mathtt{1},\mathtt{1}) \text{ or } x = \mathtt{A} \text{ or } y = \mathtt{A} \text{ or } z = \mathtt{A} \\ \mathtt{1} & \text{if } (x,y,z) = (\mathtt{0},\mathtt{0},\mathtt{0}) \\ \mathtt{A} & \text{otherwise.} \end{cases}$$

**Theorem 2.** *The rule $f_{\mathrm{A}}$ solves the relaxed version of the global synchronisation problem in the sense that any configuration in $\mathcal{E}_n = \{\mathtt{0},\mathtt{1}\}^{\mathcal{L}}$ is synchronised to* $\mathbf{0}$ *or* $\mathbf{1}$.

*Proof.* To show that $f_{\mathrm{A}}$ solves the relaxed version the problem, let us consider the rule $g : {Q'_2}^5 \to Q'_2$ defined with

$$\forall (a,b,c,d,e) \in {Q'_2}^5, g(a,b,c,d,e) = f_{\mathrm{A}}\big(f_{\mathrm{A}}(a,b,c), f_{\mathrm{A}}(b,c,d), f_{\mathrm{A}}(c,d,e)\big)$$

In other words, we have $g = f_{\mathrm{A}} \circ f_{\mathrm{A}}$ in the usual sense of the composition of two cellular automata. Then, it can be verified that $g$ is such that:

$$g(a,b,c,d,e) = \begin{cases} \mathtt{1} & \text{if } (a,b,c,d,e) = (\mathtt{0},\mathtt{0},\mathtt{0},\mathtt{0},\mathtt{0}) \\ \mathtt{0} & \text{otherwise.} \end{cases}$$

Clearly, the global rule $G$ associated to $g$ possesses only two fixed points: $\mathtt{0}^{\mathcal{L}}$ and $\mathtt{1}^{\mathcal{L}}$. Moreover, $G$ verifies $G^{\lfloor n/2 \rfloor}(x) = \mathtt{0}^{\mathcal{L}}$ for each initial configuration $x \in \mathcal{E}_n \setminus \{\mathbf{1}\}$. (Only initial conditions that do not contain an `A` need to be considered.) As we have $G = F_{\mathrm{A}} \circ F_{\mathrm{A}}$, it directly follows that $F_{\mathrm{A}}$ synchronises every initial condition in at most $n$ steps. □ □

The proof above may seem quite cryptic. To understand how $g$ operates, let us come back to the original rule $f_{\mathrm{A}}$. Remark that in the definition of $f_{\mathrm{A}}$, the state `A` is used to "mark" the regions where `0`-cells "meet" `1`-cells. Indeed, if the quintuplet $(a, b, c, d, e)$ is equal to $\mathtt{0}^5$ or to $\mathtt{1}^5$, then the central cell will blink twice. If the quintuplet $(a, b, c, d, e)$ contains both states `0` and `1`, then the central cell becomes an `A` at the next time step, and then a `0` at the following time step. The state `A` can thus appear only at odd time steps and this state is used only to synchronise locally the configuration on the state `0`. The evolution below shows an example of how the rule synchronises a given initial configuration in four steps. (Time runs from bottom to top.)

| | |
|---|---|
| `00000000000000000000000` | $t = 4$ |
| `11111111111111111AAAA11` | $t = 3$ |
| `00000000000000000011000` | $t = 2$ |
| `11AA0AAAAA1AAA1AA0000AA` | $t = 1$ |
| `00011100100010001111110` | $t = 0$ |

Note that Richard has presented a construction which uses 25 auxiliary states to achieve the synchronisation of rings of odd size [Ric17]. In fact, although more costly in terms of states, this rule has the advantage to solve the problem for any initial condition and not only for the initial conditions which do not contain an auxiliary state. We thus need to mind the fact that small variations in the problem can lead to the construction of quite different solutions.

## 3.3 General Case

The previous construction can readily be extended for any number of states: one simply needs to use more auxiliary states. If we have $k$ states, then we use the alphabet $Q'_k = Q_k \cup \{\mathtt{A}_i, i \in \{1, \ldots, k-1\}\}$. The local rule $f_{\mathrm{A}} : {Q'_k}^3 \to Q'_k$ then reads:

$$f_{\mathrm{A}}(x, y, z) = \begin{cases} q + 1 & \text{if } (x, y, z) = (q, q, q) \text{ for all } q \in Q_k, \\ \mathtt{A}_i & \text{if } \mathtt{A}_{i-1} \in \{x, y, z\} \text{ for all } i \in \{2, \ldots, k-1\}, \\ \mathtt{0} & \text{if } \mathtt{A}_{k-1} \in \{x, y, z\}, \\ \mathtt{A}_1 & \text{otherwise}^3. \end{cases}$$

In words, the rule is composed of four rules (see Fig. 3):

1. the cyclicity condition for homogeneous neighbourhoods,

2. the auxiliary states "contaminate" their neighbours and shift their state,

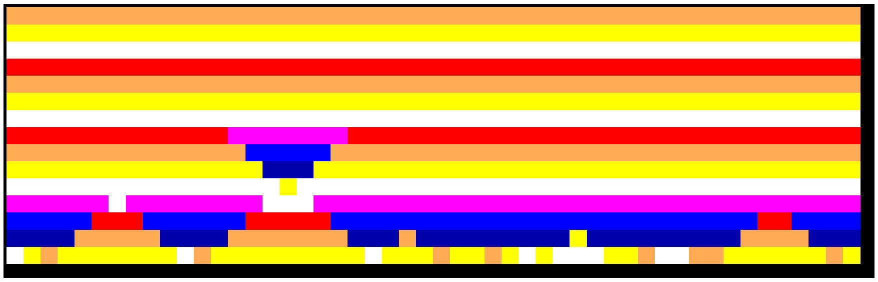

Figure 3: Space-time diagram of a rule that achieves the synchronisation for all the initial conditions. There are $k = 4$ states, associated to the following colours: white, yellow, orange, and red. The auxiliary states $\texttt{A}_1$, $\texttt{A}_2$ and $\texttt{A}_3$ are respectively in dark blue, blue and purple.

3. once the last auxiliary state has been reached, the cell returns to `0`,
4. any cell which has a neighbourhood that is not homogeneous and does not contain an auxiliary state turns to the first auxiliary state $\texttt{A}_1$.

Using these rules, we can ensure the synchronisation of any configuration with the two following properties:

- An auxiliary state $\texttt{A}_i$ can only appear at a time $t$ such that: $t \mod k = i$.
- If a cell is in state `0` in a given configuration $x$, then all the cells at distance less or equal than $k$ are also in state `0` in $F^k(x)$, that is, after $k$ applications of the global rule.

Indeed, for $i \in \{1, \dots, k-1\}$, it is clear by recurrence that if a cell $c$ is in state `0` in $x$, then, in $F^i(x)$, the cells of the set $\{c + d, d \in \{-i, \dots, i\}\}$ are either in state $i$ or in state $\texttt{A}_i$.

Note that when $k$ is large, one can use a logarithmic number of auxiliary states (with $k$) by implementing an internal binary counter. However, it is an open problem to know whether a perfect solution can be designed with a number of auxiliary states that does not depend on $k$.

### 3.4 A rule which synchronises "almost all" initial conditions

We now present a construction that works for any number of states, but which achieves the synchronisation only if the initial condition contains at least one cell in state `0` (the choice of state is of course arbitrary).

We extend the set $Q_k$ with three auxiliary states: the `I` and `E` states respectively denote the *interior* and *exterior* markers of the extending zones, and the state `D` is used for "deactivating" some cells. The state set is thus:

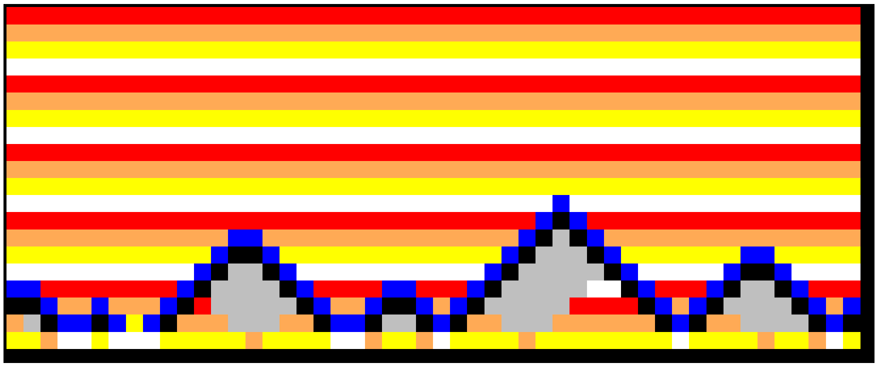

Figure 4: Space-time diagram of a rule that achieves the synchronisation for almost all the initial conditions. There are $k = 4$ states, associated to the following colours: white, yellow, orange, and red. The auxiliary states $\mathtt{E}$, $\mathtt{I}$, and $\mathtt{D}$ are respectively in black, blue and grey. (Colour online)

$Q'_k = Q_k \cup \{\mathtt{I}, \mathtt{E}, \mathtt{D}\}$. The idea is to select one particular state (state $\mathtt{0}$) and mark the intervals of consecutive cells which have this state. These intervals will be made to grow and "eat" the passive cells. In the intervals, the evolution follows the normal cycle of $Q_k$. Moreover, when two growing intervals meet, they simply merge and their frontiers disappear. At the end it is expected that all the markers of the intervals and the passive cells have disappeared and the whole systems oscillates in $Q_k$ (see Fig. 4).

These rules are described as follows. First let us examine the behaviour of non-auxiliary cells.

- The rule follows the cyclicity condition: $\forall q \in Q_k,\ f(q, q, q) = q + 1$.
- The interior of the intervals of $\mathtt{0}$'s are marked with $\mathtt{I}$'s: $f(\mathtt{0}, \mathtt{0}, q) = f(q, \mathtt{0}, \mathtt{0}) = f(q, \mathtt{0}, q') = \mathtt{I}$ for $q, q' \in Q_k \setminus \mathtt{0}$.
- Their exterior is marked with $\mathtt{E}$'s: $\forall q, q' \in Q_k \setminus \mathtt{0}, f(\mathtt{0}, q, q') = f(q', q, \mathtt{0}) = \mathtt{E}$.
- All the cells which are not $\mathtt{0}$'s and do not see any $\mathtt{0}$ are deactivated: $\forall q_1, q_2, q_3 \in Q_k \setminus \mathtt{0}, f(q_1, q_2, q_3) = \mathtt{D}$.

The auxiliary states react as follows.

- Any non-auxiliary cell that sees an $\mathtt{E}$ turns to an $\mathtt{E}$: $\forall q, q' \in Q_k,\ f(\mathtt{E}, q, \mathtt{E}) = f(\mathtt{E}, q, q') = f(q', q, \mathtt{E}) = \mathtt{E}$. The exterior markers always become interior markers: $\forall q, q' \in Q_k,\ f(q, \mathtt{E}, q') = \mathtt{I}$.
- The interior markers normally have at least one neighbour with a state in $Q_k$, unless they are surrounded by two $\mathtt{E}$ cells. The interior cells thus synchronise with their neighbours: $\forall q \in Q_k, q_{\mathtt{A}} \in \{\mathtt{I}, \mathtt{E}\}; f(q, \mathtt{I}, q) =$

$f(q, \mathtt{I}, q_{\mathtt{A}}) = f(q_{\mathtt{A}}, \mathtt{I}, q) = q + 1$. Note that by construction, the interior cells should never see any deactivated cell, nor should they see two different states from $Q_k$. Moreover, if a cell $c$ is in state I and surrounded by two cells in state E, it implies that $c$ was in state 0 in the previous time step. It thus turns to state 2 at the next time step: $f(\mathtt{E}, \mathtt{I}, \mathtt{E}) = 2$.

- The behaviour of deactivated cells is quite simple: they remain in their state until they are "eaten" by an exterior cell; we thus write: $\forall q, q' \in Q_k \cup \{\mathtt{D}, \mathtt{I}\}$, $f(q, \mathtt{D}, q') = \mathtt{D}$ and: $\forall q \in Q_k \cup \{\mathtt{E}, \mathtt{I}, \mathtt{D}\}$, $f(q, \mathtt{D}, \mathtt{E}) = f(\mathtt{E}, \mathtt{D}, q) = \mathtt{E}$.

Let us prove informally that the synchronisation occurs for an initial condition of ${Q_k}^{\mathcal{L}}$ which contains at least one cell in state 0.

As it can be seen on Fig. 4, this rule is constructed in order to ensure that a state $q \in Q_k$ can only appear in a time $t$ such that $t \mod k = q$. Let us then call the "inner intervals" the consecutive set of cells which have a IE pattern on their right and a EI pattern on their left. We claim that as long as the system have not synchronised, the size of the inner intervals strictly increases. Indeed, when an inner interval does not meet another interval, its size increases by 2 cells at each time step. When two inner intervals meet, they merge: first the two E's disappear and then the two I's. This phenomenon occurs in two or three steps, depending on the parity of the number of cells which separate two inner intervals. This is illustrated on the example of Fig. 4, where the two types of merging events occur. At the end, it is expected that all the inner intervals have merged, which gives a synchronised configuration. The initial conditions which do not contain at least one 0 will evolve to an all-D fixed point.

The advantage of the constructions presented above is that they can be extended to higher dimensions quite directly. Indeed, as these rules only use "dimension-independent" notions such as interior, exterior, homogeneity of the neighbourhood, etc., they can be transposed to various spaces.

However, the rules given above rely on an artificial symmetry breaking: the desired synchronisation phase, that is, the time modulo $k$ at which each synchronised state appears, is already known in advance. Another way to break the symmetry is to use fixed boundary conditions (see Appendix Sec. 5.1), but here too the source of the symmetry breaking is quite artificial. In the next section, we consider a more satisfying framework where randomness is taken as a resource for inducing the synchronisation, with the advantage that we will propose colour-blind rules, or, with other words, we will not make any distinction between the states.

# 4 "Perfect" stochastic solutions

We now propose to examine the situation of the synchronisation problem in the universe of stochastic rules. In fact, if we allow randomness in the transitions of the rule, it becomes difficult *not* to solve the problem! For the sake of simplicity we restrict our study to the nearest-neighbour interactions: $\mathcal{N} = \{-1, 0, 1\}$.

## 4.1 The binary case

We first examine the binary case for nearest neighbours (probabilistic ECA). We consider a local transition function $\phi : {Q_2}^3 \to [0, 1]$, which associates to each neighbourhood state its probability to be updated to $\mathtt{1}$.

Formally, starting from a configuration $x$, the system can be described by a *stochastic process* $(x^t)_{t\in\mathbb{N}}$. The sequence $(x^t)$ now denotes a sequence of *random variables* defined recursively with: $x^0 = x$ (with probability 1) and

$$\forall t \in \mathbb{N},\, \forall i \in \mathcal{L},\, x_i^{t+1} = \mathcal{B}(\phi(x_{i-1}^t, x_i^t, x_{i+1}^t)), \tag{2}$$

where $\mathcal{B}(p)$ is the Bernoulli random variable which equals $\mathtt{1}$ with probability $p$ and $\mathtt{0}$ with probability $1-p$. Note that, strictly speaking, the definition above is more a characterisation than a definition and that a "proper" definition would require the use of tools from measure theory (see e.g. Ref. [MM14]).

We now need to redefine what it means to solve the problem perfectly. The blinking condition is easily translated to $\phi(\mathtt{0}, \mathtt{0}, \mathtt{0}) = 1$ and $\phi(\mathtt{1}, \mathtt{1}, \mathtt{1}) = 0$. For a rule which satisfies the blinking condition and for a given configuration $x \in \mathcal{E}_n$, we define the synchronisation time $T(x)$ as the random variable which corresponds to the number of steps needed to attain one of the two homogeneous configurations: $T(x) = \min\{t, x^t \in \{\mathbf{0}, \mathbf{1}\}\}$. The average synchronisation time of $x$ is the expectation of $T(x)$, denoted by $\mathbb{E}\{T(x)\}$. For a given size $n$, we define the worst expected synchronisation time (WEST) and the expected average synchronisation time (EAST) as:

$$\text{WEST}(n) \quad = \max_{x\in\mathcal{E}_n} \mathbb{E}\{T(x)\}, \tag{3}$$

$$\text{EAST}(n) \quad = \tfrac{1}{2^n} \textstyle\sum_{x\in\mathcal{E}_n} \mathbb{E}\{T(x)\}. \tag{4}$$

In this probabilistic framework, we say that $f$ *synchronises* the size $n$ if $\text{WEST}(n)$ is finite. Clearly, this is equivalent to having a finite $\text{EAST}(n)$. By extension, $f$ is a solution to the global synchronisation problem if $f$ synchronises all sizes $n \in \mathbb{N}$.

Let us now examine how to build a solution. For a function $\phi$, we introduce the variables: $p_0 = \phi(\mathtt{0}, \mathtt{0}, \mathtt{0})$, $p_1 = \phi(\mathtt{0}, \mathtt{0}, \mathtt{1})$, ..., $p_7 = \phi(\mathtt{1}, \mathtt{1}, \mathtt{1})$.

**Theorem 3.** *Let $\phi$ be a probabilistic ECA such that $p_0 = 1$, $p_7 = 0$ and $\forall i \in \{1, \ldots, 6\}, p_i \in ]0, 1[$, then $\phi$ is a solution to the global synchronisation problem.*

*Proof.* The evolution of the system is described by the stochastic process $(x^t)_{t\in\mathbb{N}}$. As this evolution only depends on the current state of the system, $(x^t)$ is a Markov chain and we denote by $\Pr\{x^{t+1} = \beta | x^t = \alpha\}$ the probability to go from a configuration $\alpha$ to a configuration $\beta$.

The states of Markov chains are of two kinds: transient and recurrent. A transient state is such that there is a non-zero probability that the system never returns to this state in its evolution; on the contrary, a recurrent state is visited an infinite number of times almost surely. In our system, the probability to go from $\mathbf{0}$ to $\mathbf{1}$ and from $\mathbf{1}$ to $\mathbf{0}$ is equal to 1, the two homogeneous configurations are thus recurrent; they correspond to the synchronised state of the system.

Now, consider an arbitrary configuration $x \in \mathcal{E}_n \setminus \{\mathbf{0}, \mathbf{1}\}$. Let us take again the deterministic global rule $F_{\mathrm{A}}$ which operates on the set of states $Q'_2 = \{\mathtt{0}, \mathtt{1}, \mathtt{A}\}$ (see Sec. 3.2 p. 9). Let $R : Q'_2 \to Q_k$ be the projection function defined by $R(\mathtt{0}) = \mathtt{0}$, $R(\mathtt{1}) = \mathtt{1}$ and $R(\mathtt{A}) = \mathtt{1}$. Let $(y^t) \in (E_{\mathrm{A}})^{\mathbb{N}}$ and $(z^t) \in (\mathcal{E}_n)^{\mathbb{N}}$ be the sequences of configurations defined by $y^t = (F_{\mathrm{A}})^t(x)$ and $z^t = R(y^t)$. Note that $y^0 = z^0 = x$. Now, one can easily verify that $(z^t)$ is such that $\Pr\{x^{t+1} = z^{t+1}, x^t = z^t\} > 0$ for any $t \in \mathbb{N}$. In words, this means that $(x^t)$, the original process defined with $\phi$, may follow exactly the process $(z^t)$ with a non-zero probability (during any finite number of steps). In fact, each time that an $\mathtt{A}$ can be created by $F_{\mathrm{A}}$, an $\mathtt{0}$ can be created by $\phi$.

By the properties of $F_{\mathrm{A}}$, we have $y^t \in \{\mathbf{0}, \mathbf{1}\}$ for all $t \geq n$ (see Sec. 3.2 p. 9), which also implies $z^t \in \{\mathbf{0}, \mathbf{1}\}$ ; as a consequence, there is a non-zero probability to go from the configuration $x$ to one of the two homogeneous configurations, but no possibility to return to $x$. This configuration is thus a *transient* state of the Markov chain; it will be synchronised in finite time almost surely. □ □

Note that the proposition above only gives *sufficient* conditions to solve the problem; moreover, it does not give any idea on the time needed to converge to a fixed point. This means in particular that the function $\mathrm{WEST}(n)$ may scale exponentially with $n$, which is not what one expects.

**Theorem 4.** *For $p \in ]0, 1/2[$, consider the probabilistic ECA $\phi$ defined with the following probability to update to a $\mathtt{1}$:*

$$\phi(x, y, z) = 1 - \big(p \cdot x + (1 - 2p) \cdot y + p \cdot z\big). \tag{5}$$

*This rule solves the synchronisation problem and we have $\mathrm{WEST}(n) \in \mathcal{O}(n^2)$, that is, its average synchronisation time scales quadratically with the ring size.*

*Proof.* The rule $\phi$ is also defined by the table below:

| $(x, y, z)$ | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| $\phi(x, y, z)$ | $1$ | $1 - p$ | $2p$ | $p$ | $1 - p$ | $1 - 2p$ | $p$ | $0$ |

Now consider the probabilistic ECA $C_{\mathrm{F}} = 1 - \phi$, that is, the stochastic rule where the probability to update to a $\mathtt{1}$ is exactly the inverse of the one defined above. Let $(x^t)$ and $(y^t)$ be the Markov chains that describe the evolution of $\phi$ and $C_{\mathrm{F}}$, respectively. Let us show that there exists a coupling between the two Markov chains, more precisely that two global applications of $\phi$ and $C_{\mathrm{F}}$ have the same effect. Let $\mathsf{Bl} : \mathcal{E}_n \times \mathbb{N} \to \mathcal{E}_n$ be the *blinking operator*, defined with: $\mathsf{Bl}(x, t) = \begin{cases} x & \text{if } t \in 2\mathbb{Z}, \\ \overline{x} & \text{if } t \in 2\mathbb{Z} + 1 \end{cases}$, where the "bar" operator represents the global inversion on configurations $\overline{x} = (\overline{x}_i)_{i \in \mathcal{L}} = (1 - x_i)_{i \in \mathcal{L}}$.

Let us now remark that if $x^0 = y^0$, then for all $t \in \mathbb{N}$, $x^t = \mathsf{BI}(y^t)$. (Note that we make here a small abuse in notation, applying BI to random variables instead of "classical" configurations.) This property can be shown by recurrence. It is true by construction at time $t = 0$; let us assume that it is true for $t = 2k$.

As we have $C = 1 - \phi$, we have $x^{t+1} = \mathsf{BI}(y^{t+1})$ by the linearity of the BI operator on all the cells. Formally, this translates to

$$\forall \alpha, \beta \in \mathcal{E}_n,\ \Pr\{y^{t+1} = \beta | y^t = \alpha\} = \Pr\{x^{t+1} = \overline{\beta} | x^t = \alpha\}. \tag{6}$$

For the next step, we only need to prove that:

$$\forall \alpha, \beta \in \mathcal{E}_n,\ \Pr\{y_i^{t+2} = \beta | y_i^{t+1} = \overline{\alpha}\} = \Pr\{x^{t+2} = \beta | x^{t+1} = \alpha\}. \tag{7}$$

This second assertion can be proved by noticing that for all $(x, y, z) \in Q^3$,

$$\phi(\overline{x}, \overline{y}, \overline{z}) = \phi(1 - x, 1 - y, 1 - z) = 1 - C_{\mathrm{F}}(1 - x, 1 - y, 1 - z) = C_{\mathrm{F}}(x, y, z).$$

In words, this means that applying $\phi$ composed with the global inversion operator is equivalent to applying $C_{\mathrm{F}}$.

The rule $C_{\mathrm{F}}$ was proposed by Fukś to solve the density classification problem with a random evolution of the system. Its dynamics was analysed in a previous work and it was shown that its worst expected convergence time (WECT) scales quadratically with $n$ [Fat13].

□ □

An open question is to find the optimal value of $p$ that minimises the WEST or the EAST. As far as the latter measure is concerned, we experimentally noticed that the optimal $p$ is close to 1/2 but slightly lower than this value [Fat13]. However, for all the values of $p$, the scaling of the synchronisation time is quadratic with the ring size. In order to construct a rule whose convergence time is linear, an idea would be to find a rule which is colour-blind and which converges to a homogeneous fixed point in linear time and then to transform it with the same operation as above. One may think about the Traffic-Majority rule for example[4]. The WECT of this rule was conjectured to be linear with the ring size [Fat13]. Unfortunately, unlike $\phi$, this rule is not colour-blind and thus does not commute with the conjugation operator. It is an open question to find a rule with a synchronisation time that scales linearly with the size of the configurations.

## 4.2 The many-state case

Let us now consider the case with more than two states. A simple way to synchronise the system is to use the *uniform-copy-and-state-shift rule*: the probability to update to a state $q' = q+1$ is proportional to the number of occurrences of $q$ in the neighbourhood.

[4] This rule consists in applying the traffic rule (ECA 184) with probability $1 - \epsilon$ and the majority rule with probability $\epsilon$. It allows one to classify the density with an arbitrary precision for $\epsilon$ small enough, at the cost of an increase in the convergence time [Fat13].

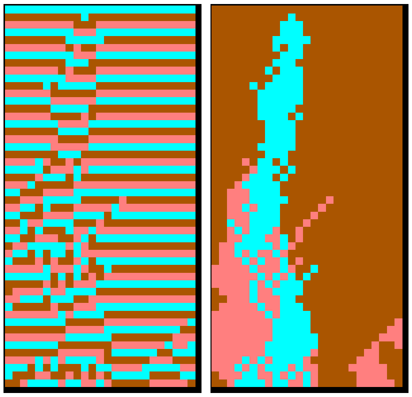

Figure 5: (left) : Space-time diagram of $\phi_k$ for $k = 3$ states. (right): the same evolution but without the applying the state-shift operator $\mathbb{S}$ to the transition function. Here, each cells simply copies uniformly at random state in the set of states of its neighbours. The three colours code the three states 0,1,2. (The exact mapping is not important since the rule is colour-blind [ST14].)

**Theorem 5.** *Let $\phi_k$ be the local stochastic function given by:*

$$\forall (x, y, z) \in {Q_k}^3, \phi_k(x, y, z) = \mathcal{U}\{x, y, z\} + 1,$$

*where $\mathcal{U}(E)$ is the random variable that draws an element uniformly in $E$. This rule solves the synchronisation problem and we have* $\mathrm{WEST}(n) \in \mathcal{O}(n^2)$*.*

Let us denote by $\langle f \rangle$ the global rule associated to a local rule $f$. The behaviour of $\langle \phi_k \rangle$ is illustrated on Fig. 5-left for $k = 3$ states. Given a neighbourhood $(x, y, z) \in {Q_k}^3$, the probability that a cell updates to state $q + 1$ is $p_q(x, y, z) = 1/3 \cdot (\delta_{q,x} + \delta_{q,y} + \delta_{q,z})$, where $\delta$ is the Kronecker symbol. To see why $\langle \phi_k \rangle$ converges in quadratic time, it is useful to remove the state-shift effect and to look simply at $\langle \mathcal{U} \rangle$, that is, the *uniform copy* rule (see Fig. 5-right). The probability that $\mathcal{U}$ updates to state $q$ is $p_q(x, y, z)$ as described above. Moreover, we have: $\forall t \in \mathbb{N}, {\langle \phi_k \rangle}^t = \mathbb{S}^t \circ {\langle \mathcal{U} \rangle}^t$. The synchronisation time of $\langle \phi_k \rangle$ is thus equal to the time of convergence of $\langle \mathcal{U} \rangle$ to a homogeneous configuration. If we could know in advance what is the "winning state", that is, the state which forms the homogeneous configuration on which $\langle \mathcal{U} \rangle$ will converge, we could project the evolution of this $k$-state rule on a 2-state rule and immediately use the result above to obtain a quadratic convergence time. Unfortunately, it is not possible to do so, at least not directly, because this would amount to condition the random variables by future events. We thus need to adapt the lemma on the quadratic convergence of martingales for this particular case, we present the proof of this theorem in the appendix.

Note that this rule can also be applied to larger neighbourhoods and on grids with higher dimensions.

# 5 Questions

Although less studied than many other inverse problems, the global synchronisation problem constitutes a rich field of investigation. The major observation has the form of a paradox: in spite of the non exitence of a perfect determinictic solution, allowing the local transition function to use randomness is sufficient to solve the problem. This illustrates the existence of a huge gap between deterministic and stochastic systems: the use of randomness gives “perfect” solutions, although the time needed to converge is subject to variation. In general, estimating the average time of convergence is a delicate operation but we have shown that there are some rules whose convergence time scales quadratically with the number of cells. By contrast, we also noticed experimentally that some settings made the system converge with an exponential time. It would be interesting to study the changes of convergence time and relate this question to the occurrence of phase transitions in probabilistic CA [Reg13].

This of course does not imply that stochastic systems always have more abilities than their deterministic counterparts: there are simply some cases when randomness acts as a useful means to break the symmetries (see also e.g. Ref.[RR18] for a recent result on the formation of discrete lines).

To explore further the deterministic rules, the synchronisation problem can also be systematically explored by using SAT solvers. Interestingly, contrarily to the statistical approaches (see e.g. Ref. [OMdCF09]), the size of the configurations we need to consider are relatively small. The preliminary results presented in the appendix (Sec. 6) can largely be improved, notably by taking advantage of the symmetries of the problem or by using the constraints expressed in Prop. 3. It is a nice challenge to find rules which achieve perfect synchronisation on larger ring sizes.

We end our discussion with a list of questions:

**Question 1.** *Does there exist a rule which synchronises an infinite number of sizes?*

**Question 2.** *What kind of analytical results could be given for rules which would synchronise almost all initial conditions without the use of auxiliary states?*

**Question 3.** *Given a neighbourhood of size $s$, how can we effectively determine the amplitude this neighbourhood, that is, the greatest value of $m$ for which all the ring sizes smaller than or equal to $m$ that can be synchronised? For $k$ states, can we give some bounds that are tighter than $k^s$?*

**Question 4.** *Given a neighbourhood of size $s$ and a maximum synchronisation time $\tau$, what is a good algorithm to find all the rules with a height less or equal to $\tau$? What is the complexity of this problem?*

**Question 5.** *Is there a stochastic rule which solves the global synchronisation problem and whose WEST scales linearly with the ring size?*

## acknowledgements

The author is grateful to Irène Marcovici, Jean Mairesse and Sukanta Das for stimulating discussions on the topic and to the anonymous reviewers for precious indications and suggestions.

# References

[BdOF13] Heather Betel, Pedro P. B. de Oliveira, and Paola Flocchini. Solving the parity problem in one-dimensional cellular automata. *Natural Computing*, 12(3):323–337, 2013.

[BFMM13] Ana Bušić, Nazim Fatès, Jean Mairesse, and Irène Marcovici. Density classification on infinite lattices and trees. *Electronic Journal of Probability*, 18(51):1–22, 2013.

[BnMdEOR13] Raimundo Briceño, Pablo Moisset de Espanés, Axel Osses, and Ivan Rapaport. Solving the density classification problem with a large diffusion and small amplification cellular automaton. *Physica D: Nonlinear Phenomena*, 261:70–80, 2013.

[BR99] Franco Bagnoli and Raúl Rechtman. Synchronization and maximum lyapunov exponents of cellular automata. *Physical Review E*, 59(2):R1307, 1999.

[DCMH95] Rajarshi Das, James P. Crutchfield, Melanie Mitchell, and James E. Hanson. Evolving globally synchronized cellular automata. In *Proceedings of 6th ICGA*, pages 336–343, San Francisco, USA, 1995. Morgan Kaufmann.

[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.

[Fat13] Nazim Fatès. Stochastic cellular automata solutions to the density classification problem - when randomness helps computing. *Theory of Computing Systems*, 53(2):223–242, 2013.

[Fat15] Nazim Fatès. Remarks on the cellular automaton global synchronisation problem. In Jarkko Kari, editor, *Proceedings of AUTOMATA 2015*, volume 9099 of *Lecture Notes in Computer Science*, pages 113–126. Springer, 2015.

[Fri16] Benjamin Friedrich. Hydrodynamic synchronization of flagellar oscillators. *The European Physical Journal Special Topics*, 225(11):2353–2368, 2016.

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

[MY14] Luidnel Maignan and Jean-Baptiste Yunès. Experimental finitization of infinite field-based generalized FSSP solution. In Jaroslaw Was, Georgios Ch. Sirakoulis, and Stefania Bandini, editors, *Proceedings of ACRI 2014 – 11th International Conference on Cellular Automata for Research and Industry, Krakow,* *Poland*, volume 8751 of *Lecture Notes in Computer Science*, pages 136–145. Springer, 2014.

[OMdCF09] Gina M.B. Oliveira, Luiz G.A. Martins, Laura B. de Carvalho, and Enrique Fynn. Some investigations about synchronization and density classification tasks in one-dimensional and two-dimensional cellular automata rule spaces. *Electronic Notes in Theoretical Computer Science*, 252:121–142, 2009.

[PRK03] Arkady Pikovsky, Michael Rosenblum, and Jürgen Kurths. *Synchronization: a universal concept in nonlinear sciences*, volume 12. Cambridge university press, 2003.

[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.

[Ric17] Gaétan Richard. On the synchronisation problem over cellular automata. In Heribert Vollmer and Brigitte Vallée, editors, *Proceedings of STACS 2017*, volume 66 of *LIPIcs*, pages 54:1–54:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 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.

[RR18] Damien Regnault and Éric Rémila. Lost in self-stabilization: A local process that aligns connected cells. *Theoretical Computer Science*, 2018. In press. Available online.

[ST14] Ville Salo and Ilkka Törmä. Color blind cellular automata. *Journal of Cellular Automata*, 9(5-6):477–509, 2014.

# Appendix 1: proofs

## 5.1 The synchronisation problem with fixed boundary conditions

Another way to solve the problem is to use fixed boundary conditions. We here present a rule which solves the problem for the binary case $k = 2$. For a set of states $Q_k$, let $f : Q^3 \to Q$ be such that $f(x, y, z) = \mathtt{1}$ if and only if $(x, y) = (\mathtt{0}, \mathtt{0})$. The ECA $f$ corresponds to rule 3 according to the classical decimal numbering scheme.

Let us consider the space of $n$ cells $\mathcal{L}' = \{0, \dots, n-1\}$, the configuration space $E_n = {Q_k}^{\mathcal{L}'}$, and define the following global rule $H : E_n \to E_n$ such that for all $x \in E_n$: $H(x)_0 = \mathtt{0}$, $H(x)_{n-1} = \mathtt{0}$ and $\forall i \in \{1, \dots, n-2\}$, $H(x)_i = f(x_{i-1}, x_i, x_{i+1})$. The actual state space on which this rule operates is $\{x \in {Q_k}^n, x_0 = \mathtt{0} \text{ and } x_{n-1} = \mathtt{0}\}$.

Figure 6 shows how this rule behaves for two cases: an arbitrary configuration and a particular configuration which maximises the number of steps before synchronisation.

**Theorem 6.** *For any $n$, rule $H$ synchronises the non-boundary cells.*

*Proof.* Let us show that $H$ achieves the synchronisation task on any initial condition $x \in E_n$. Let $(x^t)_{t \in \mathbb{N}}$ be the sequence of configurations which results from the evolution of $x$: $x^0 = x$ and $x^{t+1} = H(x^t)$.

The synchronisation occurs from left to right: after two time steps, we can guarantee that at least one more cell is synchronised, that is, it has the same state as the cell on its left. This property is formalised with:

$$\forall x \in E_n,\ \forall t \in \mathbb{N},\ \forall i\{1, \dots, t+1\}, x_i^{2t} = x_1 \text{ and } x_i^{2t+1} = 1 - x_1.$$

We can prove this property by recurrence on $t$. By definition we have $x_1^0 = x_1$. To show that $x_1^1 = 1 - x_1$, it is sufficient to notice that $f(\mathtt{0}, \mathtt{0}, \cdot) = \mathtt{1}$ (rule A) and $f(\mathtt{0}, \mathtt{1}, \cdot) = \mathtt{0}$ (rule B). We can remark that this also shows that the leftmost (non-boundary) cell continuously blinks: $\forall t \in \mathbb{N}$, $x_1^{t+1} = 1 - x_1^t$.

Now, let us assume that we have $x \in E_n$ and $\tau \in \mathbb{N}$ such that : $\forall i \le \tau, x_i = \mathtt{0}$ and let us examine $y = F(x)$, the image of $x$. Then, by the use of rule A above, $y$ verifies: $\forall i \le \tau, y_i = \mathtt{1}$. If we take $x$ such that : $\forall i \le \tau, x_i = \mathtt{1}$, then, by noticing that $f(\mathtt{1}, \cdot, \cdot) = \mathtt{0}$, we have $y_{\tau+1} = \mathtt{0}$ and thus $\forall i \le \tau + 1, y_i = \mathtt{0}$, which proves that the cell at the right of $\tau$ has the same state as the cells to its left. In words, it means that the synchronisation extends by (at least) one cell to the right when the left part of the configuration has been set to `0` and simply blinks when this left part has the state `1`. This can be clearly seen on the space-time diagrams showing the evolution of the rule (see Fig. 6 and Fig. 7).

□

## 5.2 Convergence in quadratic time of the multi-state

We now present an adaptation of Lemma 1 in Ref. [Fat13] to show the quadratic convergence of the $k$-state uniform copy rule.

**Lemma 2.** *Let $n$ and $k$ be two positive integers. Consider $k$ stochastic processes $(X_1^t), \dots, (X_k^t)$, with values in $\{0, \dots, n\}$. Let $F_t$ be a filtration adapted to these processes. Informally, this means that the only information that we have at time $t$ is the value taken by this process from the initial time to the time $t$.*

*We define $T = \min\{t \in \mathbb{N}; \exists i \in \{1, \dots, k\}, X_i^t = n\}$ as the absorbing time of the process.*

*If the processes are martingales, that is, each $(X_i^t)$ verifies:*

$$\forall i \in \{1, \dots, k\}, \forall t \in \mathbb{N}; \mathbb{E}\{X_i^{t+1} | \mathcal{F}_t\} = X_i^t,$$

*and if there exists a positive constant* $\epsilon$ *such that:*
$\forall t < T, \exists i \in \{1, \dots, k\}; \mathbb{E}\{(X_i^{t+1} - X_i^t)^2 | \mathcal{F}_t\} \geq \epsilon$, *then*

$$\mathbb{E}\{T\} \leq n^2/\epsilon$$

*Proof.* From the fact that the processes are martingales, the application of the optional stopping theorem gives: $\mathbb{E}\{X_i^T\} = \mathbb{E}\{X_i^0\}$. Let us now follow the evolution of $Y_t = \epsilon t - \sum_{i=1}^k (X_i^t)^2$. We write:

$$\mathbb{E}\{(Y_{t+1} - Y_t|\mathcal{F}_t\} = \epsilon - \sum_{i=1}^{k} \mathbb{E}\{(X_i^{t+1})^2 - (X_i^t)^2\} \tag{8}$$

Let us introduce the notation, for all $i$, $X_i^{t+1} = X_i^t + \Delta X_i^t$. This yields:

$$\begin{aligned} \mathbb{E}\{(X_i^{t+1})^2 - (X_i^t)^2 | \mathcal{F}_t\} &= \mathbb{E}\{2\Delta X_i^t \cdot X_i^t + \Delta X_i^t)^2 | \mathcal{F}_t\} \\ &= 2X_i^t \cdot \mathbb{E}\{\Delta X_i^t | \mathcal{F}_t\} + \mathbb{E}\{(\Delta X_i^t)^2 | \mathcal{F}_t\} \\ &= \mathbb{E}\{(\Delta X_i^t)^2 | \mathcal{F}_t\}, \end{aligned}$$

by using the linearity of the expectation and the hypothesis $\mathbb{E}\{\Delta X_i^t | \mathcal{F}_t\} = 0$. From the second hypothesis of the lemma, there exists a value of $i$ for which $\mathbb{E}\{(\Delta X_i^t)^2 | \mathcal{F}_t\} > 0$. This implies that for all $t \leq T$, $\mathbb{E}\{Y_t\} \leq \mathbb{E}\{Y_0\}$, that is, $(Y_t)$ is a supermartingale.

On the one hand, we write $\mathbb{E}\{Y_T\} \leq \mathbb{E}\{Y_0\}$, and, on the other hand, we apply the optional stopping theorem, we write: $\mathbb{E}\{Y_T\} = -n^2$. The estimation of the upper bound of $\mathbb{E}\{Y_T\}$ gives: $\mathbb{E}\{Y_0\} = -\sum_{i=1}^k \mathbb{E}\{(X_0)^2\} \leq 0$, which yields the result: $\epsilon \mathbb{E}\{T\} \leq n^2$.

□ □

*of Theorem 5.* Recall that we want to apply the lemma to the uniform copy rule $\langle \mathcal{U} \rangle$ to prove that it converges to a homogeneous fixed point in quadratic expected time. The values $k$ and $n$ of the lemma, respectively denote the number of states and number of cells of our cellular automaton. If $(x^t)_{t \in \mathbb{N}}$ is the sequence random variables that describe the evolution of the system, the $k$ processes are defined with: $\forall q \in Q_k, X_q^t = |x^t|_q = |\{c \in \mathcal{L}; x_c^t = q\}|$, that is, each process $(X_q^t)$ counts the number of cells in state $q$.

To verify the first hypothesis of the lemma, we estimate for each cell the average variation of each state. Recall that for a random configuration $x^t$ and its image $x^{t+1}$, we have, for each cell $c \in \mathcal{L}$ and for all possible neighbourhoods $\alpha, \beta, \gamma \in Q_k$:

$$\mathbb{E}\{x_c^{t+1} = q | x_{c-1}^t = \alpha, x_c^t = \beta, x_{c+1}^t = \gamma\} = 1/3 \cdot (\delta_{q,\alpha} + \delta_{q,\beta} + \delta_{q,\gamma}).$$

From this, it directly follows that for all $q \in Q_k$:

$$\begin{aligned} \mathbb{E}\{X_q^{t+1} - X_q^t\} &= \sum_{c \in \mathcal{L}} \mathbb{E}\{x_c^{t+1} = q\} - \mathbb{E}\{x_c^t = q\}, \\ &= \sum_{c \in \mathcal{L}} \frac{1}{3} \cdot (\delta_{q,x_{c-1}^t} + \delta_{q,x_c^t} + \delta_{q,x_{c+1}^t}) - \delta_{q,x_c^t} = 0 \end{aligned}$$

by remarking that the four sums annihilate.

To verify the second hypothesis, we only need to remark that as long as there is one cell which has a non-homogeneous neighbourhood, there are at least three cells which can change their state with a probability 1/3 or to 2/3. This means that we have for $t < T$, $\exists q \in Q_k, \mathbb{E}\{(\Delta X_i^t)^2 | \mathcal{F}_t\} \geq 1/3$, which allows us to apply the lemma with $\epsilon = 1/3$. □ □

# 6 Appendix 2: Exploring some limits with a SAT solver

In Sec. 3.1, we established the existence of a limit in the interval of sizes that can be synchronised: a rule of size $s$ cannot synchronize all the sizes from 1 to $k^s$. However, this result does not tell us what are the actual limits nor does it let us know how to construct a rule that synchronises a given size. We now propose to do an "automated exploration" of these questions for the binary case $k = 2$.

In short, the idea is to model the transition table of a CA rule as a sequence of Boolean variables and then to express a Boolean formula to state that a configuration is synchronised. For a fixed neighbourhood size, we proceed by progressively increasing the size of the configurations that need to be synchronised. If we are not stop by the resources used for the computation, we should reach a point where we find that the problem is no longer satisfiable. This point would allow us to find the maximum synchronisation length of a given neighbourhood size.

## 6.1 General SAT formulation

A SAT problem consists in finding, given Boolean formula, whether there exists an assignment of variables that satisfies this formula. A *clause* is a disjunction of literals (e.g., $a \vee \neg b \vee c$). Our SAT problem will be formulated in a conjunctive normal form (CNF): it is a conjunction of clauses. In the sequel, we call a *CNF formula* any of such conjunction of clauses.

For a neighbourhood of size $s$, let $M = 2^s$ be the number of transitions[5] of this neighbourhood. We introduce $M$ Boolean variables $t_0, \dots, t_{M-1}$ to encode the transitions of a rule $f$; the convention is that $t_i$ is true if and only if $f\big(\mathfrak{b}_s(i), \dots, \mathfrak{b}_1(i)\big) = 1$. For an initial condition $x \in \mathcal{E}_n$, we also introduce $(\tau + 1) \cdot n$ additional variables, denoted by $(\sigma(t,i))_{t \in \{0,\dots,\tau\}, i \in \{0,\dots,n-1\}}$, which correspond to the values $(x_i^t) \in Q_k$ taken by the cells in the evolution of $x$.

### 6.1.1 Blinking condition

The blinking condition $f(q, \dots, q) = \overline{q}$ is simply expressed by a CNF formula with two atomic clauses: $\mathrm{F}_{\mathrm{bl}} = t_0 \wedge \neg t_{M-1}$.

[5] A transition is a t-uple of size $s$ which is given as an input to the local rule.

### 6.1.2 Initial state

Let us now see how to encode the states of an initial condition $x \in \mathcal{E}_n$ in a formula. The operation simply consists in "translating" the initial condition $x$ into the CNF formula:

$$\mathrm{F}_{\mathrm{ic}}(x) = \bigwedge_{i \in \{0,\ldots,n-1\}} \mathbb{H}\{\sigma(0,i), x_i^0\}. \tag{9}$$

where $\mathbb{H}\{V, q\}$ is a function which is equal to the variable $V$ if $q = \mathtt{1}$ and to the variable $\neg V$ otherwise.

For instance if we have $x = \mathtt{0011}$ as an initial condition, the associated formula will be:

$$\mathrm{F}_{\mathrm{ic}}(x) = \neg\sigma(0,0) \wedge \neg\sigma(0,1) \wedge \sigma(0,2) \wedge \sigma(0,3). \tag{10}$$

### 6.1.3 Synchronisation condition

Let $\tau$ be the maximum number of time steps to achieve the synchronisation. To express the condition that $x$ is synchronised in at most $\tau$ steps, we can write $x^\tau \in \{\mathbf{0}, \mathbf{1}\}$. Unfortunately, this cannot be translated in a straightforward way into a CNF formula. Indeed, if we write:

$$\mathrm{F}_{\mathrm{sc}}(x,\tau) = \left( \bigwedge_{i \in \{0,\ldots,n-1\}} \sigma(\tau,i) \right) \bigvee \left( \bigwedge_{i \in \{0,\ldots,n-1\}} \neg\sigma(\tau,i) \right), \tag{11}$$

we need to distribute the `and` operator over the `or` to obtain a CNF. This is why we prefer to formulate this condition as:

$$\mathrm{F}_{\mathrm{sc}}(x,\tau) = \bigwedge_{i \in \{0,\ldots,n-2\}} \sigma(\tau,i) = \sigma(\tau,i+1), \tag{12}$$

which simply represents the fact that all the states of $x^\tau$ are equal. By noting that $a = b$ is equivalent to $(a \vee \neg b) \wedge (\neg a \vee b)$, $\mathrm{F}_{\mathrm{sc}}$ becomes:

$$\mathrm{F}_{\mathrm{sc}}(x,\tau) = \bigwedge_{i \in \{0,\ldots,n-2\}} (\sigma(\tau,i) \vee \neg\sigma(\tau,i+1)) \wedge (\neg\sigma(\tau,i) \vee \sigma(\tau,i+1)). \tag{13}$$

### 6.1.4 Consistency conditions

We now need to write a CNF formula for the condition: $x^{t+1} = F(x^t)$ for $t \in \{0,\ldots,\tau-1\}$. We call this formulation the "*consistency condition*", as it expresses the fact that a given Boolean formula is consistent with the evolution of the cellular automaton. We now give a precise description of this CNF formula. In order to ease the notations, let us detail this operation for the specific case of $\mathcal{N} = \{-1,0,1\}$; it is easy to generalise it to other neighbourhoods. Locally, our condition is expressed by

$$\forall i \in \mathcal{L},\, x_i^{t+1} = f(x_{i-1}^t, x_i^t, x_{i+1}^t) \tag{14}$$

which is translated as:

$$\forall i \in \{0, \ldots, n-1\}, \varphi(\sigma(t+1,i), \sigma(t,i^-), \sigma(t,i), \sigma(t,i^+)) \tag{15}$$

where $i^- = (i-1) \mod n$ and $i^+ = (i+1) \mod n$, and where $\varphi$ is a function that remains to be found.

Our goal is to find $\varphi$ such that $\varphi(y', x, y, z)$ is a CNF formula that expresses that $y'$ is the result of transition $f(x,y,z)$. In a usual programming environment, one would need simply to calculate $i = x + 2y + 4z$ and then to read the value of $t_i$ and to assign it to $y'$. Unfortunately, there is no direct way of "coding" these operations in a SAT formula. We thus need to enumerate all the possible values for the variables $y', x, y, z$ and then write a consistency condition that expresses that $y'$ equals $t_i$ where $i$ is the index which corresponds to the transition $(x, y, z)$.

For example, if we take transition $f(0,1,1) = 1$, we have $i = 3$ and we write the formula with five clauses:

$$\varphi_3 = \neg x \wedge y \wedge z \wedge t_3 \wedge (y' \vee \neg t_3) \wedge (\neg y' \vee t_3), \tag{16}$$

where the two last clauses stand for $y' = t_3$.

We thus decompose $\varphi$ as: $\varphi = \bigvee_{\lambda \in \{0,\ldots,7\}} \varphi_\lambda$ with:

$$\begin{aligned} \varphi_\lambda(y', x, y, z) = \quad & \mathbb{H}\{x, \lambda_1\} \wedge \mathbb{H}\{y, \lambda_2\} \wedge \mathbb{H}\{z, \lambda_3\} \wedge \\ & \mathbb{H}\{t_\lambda, f(x,y,z)\} \wedge (y' \vee \neg t_\lambda) \wedge (\neg y' \vee t_\lambda), \end{aligned} \tag{17}$$

where $\lambda_i = \mathfrak{b}_i(\lambda)$ is the value of the $i$-th bit of the binary representation of $\lambda$. By distributing the `and` operator over the `or`, $\varphi_\lambda$ becomes:

$$\begin{aligned} \varphi_\lambda = \quad & (\mathbb{H}\{x, \overline{\lambda_1}\} \vee \mathbb{H}\{y, \overline{\lambda_2}\} \vee \mathbb{H}\{z, \overline{\lambda_3}\} \vee y' \vee \neg t_\lambda) \bigwedge \\ & (\mathbb{H}\{x, \overline{\lambda_1}\} \vee \mathbb{H}\{y, \overline{\lambda_2}\} \vee \mathbb{H}\{z, \overline{\lambda_3}\} \vee \neg y' \vee t_\lambda). \end{aligned} \tag{18}$$

Each elementary transition of a given cell at a given time step is thus encoded with a formula $\varphi$ which contains $2M = 2^{s+1} = 16$ clauses. As there are $n\tau$ such elementary conditions, the consistency condition is given by $\mathrm{F_e}$ with $2^{s+1} \cdot n\tau$ clauses:

$$\mathrm{F_e}(x, \tau) = \bigwedge_{\substack{t \in \{0,\ldots,\tau-1\}, \\ i \in \{0,\ldots,n-1\}}} \varphi\big(\sigma(t+1,i), \sigma(t,i^-), \sigma(t,i), \sigma(t,i^+)\big). \tag{19}$$

### 6.1.5 Combining initial conditions

The last step that remains is to combine various initial conditions in order to find out if the problem is or is not solvable for a given set of initial conditions.

We proceed iteratively by increasing the size of the initial conditions to synchronise. From Prop. 2 in Sec. 2.3, we know that we do not need to consider all the initial conditions: for each size $n$, it is sufficient to select only *one* initial condition in each possible set of rotations. We say that a set of configurations is *representative* if the rotations of its members form a partition of the configuration space. Formally, let us denote by $\chi(n)$ the set of representative conditions; we write:

$$\chi(n) = \{X \subset \mathcal{E}_n, \forall x, y \in X, [x] \cap [y] = \emptyset, \bigcup_{x \in X} [x] = \mathcal{E}_n\}. \tag{20}$$

The following table shows the growth of these sets[6]:

| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $\|\chi(n)\|$ | 2 | 3 | 4 | 6 | 8 | 14 | 20 | 36 | 60 |

To sum up, if we fix a set of ring sizes $\mathcal{S} = \{n_1, \ldots, n_s\}$, and a maximum synchronisation time $\tau$, we construct a sequence of *sets* of configurations: $X_1, \ldots, X_s$ such that $X_i \in \chi(n_i)$ and build the *general formula*:

$$\mathrm{F}_{\mathrm{synch}}(\mathcal{S}, \tau) = \mathrm{F}_{\mathrm{bl}} \bigwedge_{i \in \{1,\ldots,s\}} \bigwedge_{x \in X_i} \mathrm{F}_{\mathrm{ic}}(x, \tau) \wedge \mathrm{F}_{\mathrm{sc}}(x, \tau) \wedge \mathrm{F}_{\mathrm{e}}(x, \tau). \tag{21}$$

This is the final formula; it expresses the fact that all the configurations of size $n \in \mathcal{S}$ are synchronised in at most $\tau$ time steps.

## 6.2 Experimental results

The use of SAT solvers is a well-explored field of research in Computer science. As we are not a specialist of these questions, we did not endeavour to optimise the search for a solution by any means. We simply used the `minisat` solver[7] and generated the formulae with our cellular automata simulation program `FiatLux`[8]. The results below describe the results obtained for an increasing neighbourhood size. The rules are grouped according to their symmetries with respect to the reflection, conjugation, and reflection-conjugation operations.

We define the *height* $h$ of a configuration $x \in \mathcal{E}_n$ as the time needed for the system to reach a homogeneous configuration. If we denote by $(x^t)$ the sequence of configurations obtained by applying $F$ on $x$ as described above:

$$h(x) = \min\{t \in \mathbb{N}, x^t \in \mathcal{H}\}. \tag{22}$$

with the convention that $h(x) = \infty$ if $(x^t)_{t \in \mathbb{N}}$ does not reach $\mathcal{H}$. Similarly, the *height* of a given configuration space $\mathcal{E}_n$ is the maximum height of the configurations of $\mathcal{E}_n$.

For a given neighbourhood size, we will try to find the rules with the greatest *amplitude*, that is, the rules which synchronise the greatest interval of the form $\{1, \ldots, m\}$ with $m \in \mathbb{N}$.

[6] It corresponds to sequence `A000031` in the *Online Encyclopedia of integer sequences.* One reads: "In music, $|\chi(n)|$ is the number of distinct classes of scales and chords in an n-note equal-tempered tuning system", see: `https://oeis.org/A000031`

[7] see: `http://minisat.se/`

[8] see `http://fiatlux.loria.fr`

### 6.2.1 ECA space ($s = 3$)

The results with the SAT solver show that the amplitude is 3: there exists no rule which is a solution for $\mathcal{S} = \{2, 3, 4\}$. For $\mathcal{S} = \{2, 3\}$, we find that:

- $(1, 127)$ have height of 1,
- $(19, 55)$ and $(9, 65, 111, 125)$ have height of 2, and
- no rule has height of 3.

We can also explore the synchronisation for other sets of ring sizes $\mathcal{S}$. For $\mathcal{S} = \{4\}$, we find that:

- $(37, 91)$ have a height of 3,
- $(25, 67, 103, 61)$ and $(45, 101, 75, 89)$ have height of 4 and,
- no rule has a height of 5.

For ECA 61, the synchronisation process is presented on Fig. 8, p. 29. Surprisingly, for $\mathcal{S} = \{5\}$, we find that $(9, 65, 111, 125)$ synchronises with a height of 5.

### 6.2.2 The $s = 4$ space

We now examine a neighbourhood with one more cell: we take $s = 4$ and $\mathcal{N} = \{-1, 0, 1, 2\}$. The space contains $2^{2^4} = 2^{16} = 65536$ rules.

By testing increasing values of $n$ and setting the maximum synchronisation time $\tau$ equal to $\chi(n)$ (see Prop. 3), we found that for $s = 4$, the amplitude is 6, that is, for $\mathcal{S} = \{2, \ldots, 7\}$, no solution is found.

For $\mathcal{S} = \{2, \ldots, 6\}$, we find that:

- 6 rules have a height of 4: (1077,21471), (4427,11639), (11893,20875),
- 6 rules have a height of 5: (1205,17461,21215,21469), (5419,11095),
- 2 rules have a height of 6: (4363,12151).

For rule 5419, the synchronisation process is presented on Fig. 9, p. 30.

### 6.2.3 The $s = 5$ space

We increase again the neighbourhood and take $\mathcal{N} = \{-2, -1, 0, 1, 2\}$. The space contains $2^{2^5} = 2^{32} \sim 4.10^{10}$ rules. At this point, we reach the limits of our approach: The CNF formula of the problem has 74768 variables and 4563060 clauses. By progressively increasing the values of $n$ and $\tau$, our best result was to find a rule which synchronises the size interval $\mathcal{S} = \{2, \ldots, 11\}$: rule 1078270911. This rule has a height of 18 (see Fig. 1, p. 5).

There are probably other rules which solve the problem for $\tau$ higher than 18, but we leave this exploration for future work. For $\mathcal{S} = \{2, \ldots, 12\}$, no solution was found for $\tau = 30$. Surprisingly, the non-satisfiability of the formula is given rapidly, which suggests that the maximum synchronisation length for $s = 5$ is equal to 12.

We also tested the SAT solver for $s = 6$, but $\tau = 12$ is sufficient to generate SAT problems that are not solved after a substancial time of computation. We thus leave the exploration of these greater spaces as a challenge for future work.

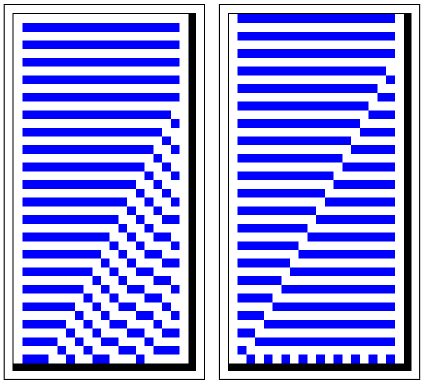

Figure 6: Space-time digram of ECA 3 with fixed boundary conditions. Blue and white squares represent cells with state `1` and `0`, respectively. Time goes from bottom to top.

⊛**00000**⊛
⊛**1111**0⊛
⊛**0000**1⊛
⊛**111**00⊛
⊛**000**10⊛
⊛**11**001⊛
⊛**00**100⊛
⊛**0**0011⊛

Figure 7: Example of an evolution of ECA 3 with fixed boundary conditions. The symbol ⊛ represents the boundary cells with state `0`. The states in bold show the synchronised part (see text). Times goes from bottom to top.

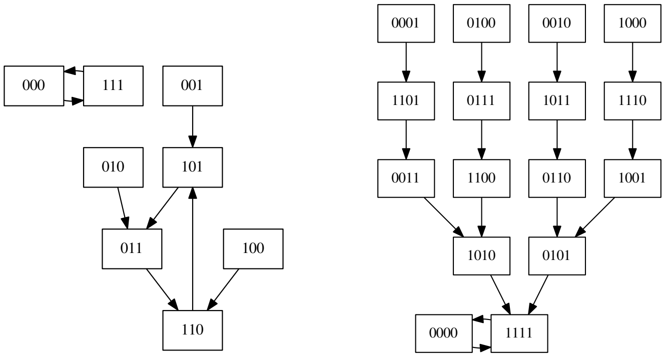

Figure 8: Transition graphs of ECA 61 for $n = 3$ and $n = 4$. An oriented link between a configuration $x$ and $y$ represents the relationship $y = F(x)$.

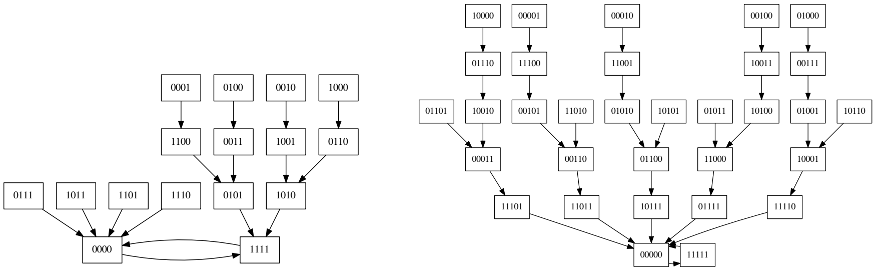

Figure 9: Neighbourhood of size $k = 4$: transition graphs of rule 5419 for $n = 4$ and $n = 5$.