Construction of μ-Limit Sets of Two-dimensional Cellular Automata

Martin Delacourt[1] **and Benjamin Hellouin de Menibus**[2]

1 **CMM, Universidad de Chile, Beauchef 851, Edificio Norte – Piso 7, Santiago, CHILE**

2 **I2M, Aix-Marseille Université, CMI, 39 rue F. Joliot Curie, 13453 Marseille Cedex 13, FRANCE**

Abstract

We prove a characterisation of μ-limit sets of two-dimensional cellular automata, extending existing results in the one-dimensional case. This sets describe the typical asymptotic behaviour of the cellular automaton, getting rid of exceptional cases, when starting from the uniform measure.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases cellular automata, dynamical systems, μ-limit sets, subshifts, measures

Digital Object Identifier 10.4230/LIPIcs.STACS.2015.262

1 Introduction

Cellular automata are discrete dynamical systems defined by a local rule, introduced in the 40s by John von Neumann [12]. They model a large variety of discrete systems and are linked with various areas of mathematics or computer science, in particular computation theory, complex systems, ergodic theory and combinatorics.

One of the main catalysts of the study of cellular automata was their surprisingly complex and organised behaviours, even when iterated on configurations with no particular structure (e.g. chosen at random). To formalise these observations, many authors tried to describe their asymptotic behaviour by considering the limit set, which is the set of configurations that can be reached after arbitrarily many steps. These sets were shown to have potentially high computational complexity [11, 1], and any nontrivial property on them is undecidable [9]. Nevertheless, the problem of characterising which subshifts can be limit sets of CA remains open.

In 2000, Kůrka and Maass argued that limit sets did not provide a good description of empirical observations and introduced instead a measure-theoretical version [10]. The idea of μ-limit sets is to choose the initial configuration at random, according to some probability measure μ, and to consider all patterns whose probability to appear does not tend to 0. In the one-dimensional case, similar results of high complexity and undecidability were found [4, 3, 6, 2]. Another approach was developed in [5], considering the limit probability measure, with similar results.

In this article, we consider the two-dimensional case and prove a characterisation of all subshifts that can be μ-limit sets of CA for μ the uniform Bernoulli measure. The method is constructive and inspired by the one-dimensional constructions in [2, 5].

32nd Symposium on Theoretical Aspects of Computer Science (STACS 2015).
Editors: Ernst W. Mayr and Nicolas Ollinger; pp. 262–274

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2 Definitions

2.1 Cellular automata on two dimensions

▶ **Definition 1** (Configurations, patterns, cylinders)**.** Let $\mathcal{A}$ be a finite alphabet. We introduce $\mathcal{A}^{\mathbb{Z}^2}$ the set of (two-dimensional) **configurations**. Denote $\mathcal{A}^*$ the set of finite **patterns**, that is, any element of $\mathcal{A}^{\mathbb{U}}$ for some $\mathbb{U} \underset{finite}{\subset} \mathbb{Z}^2$ (denote $\mathbb{U} = \mathrm{supp}(u)$ the **support** of the pattern u). Such a pattern is said to be square or rectangular if its support is.

Given $u \in \mathcal{A}^*$ and $i, j \in \mathbb{Z}^2$, define the cylinder $[u]_{i,j} = \{x \in \mathcal{A}^{\mathbb{Z}^2} \ \ x_{(i,j)+supp(u)} = u\}$.

Endowed with the product topology, $\mathcal{A}^{\mathbb{Z}^2}$ is a compact and metrisable space. A distance inducing this topology is:

$$\forall x, y \in \mathcal{A}^{\mathbb{Z}^2}, d_C(x,y) = 2^{-\Delta(x,y)} \qquad \text{where } \Delta(x,y) = \min\{|i| + |j| \mid i, j \in \mathbb{Z}^2, x_{i,j} \neq y_{i,j}\}$$

The **frequency** of a pattern $u \in \mathcal{A}^*$ in another pattern $v \in \mathcal{A}^*$ is defined as:

$$\mathrm{Freq}(u,v) = \frac{\#\left\{(i,j) \in \mathrm{supp}(v) : \begin{array}{c}(i,j) + \mathrm{supp}(u) \subseteq \mathrm{supp}(v) \\ v_{(i,j)+\mathrm{supp}(u)} = u\end{array}\right\}}{\#\left\{(i,j) \in \mathrm{supp}(v) : \ (i,j) + \mathrm{supp}(u) \subseteq \mathrm{supp}(v)\right\}} \quad , 0 \text{ if it is undefined.}$$

▶ **Definition 2** (Shift actions)**.** Define the two shifts actions $\sigma_\uparrow, \sigma_\rightarrow : \mathcal{A}^{\mathbb{Z}^2} \to \mathcal{A}^{\mathbb{Z}^2}$ by:

$$\forall x \in \mathcal{A}^{\mathbb{Z}^2}, i, j \in \mathbb{Z}^2, \qquad \sigma_\rightarrow(x)_{i,j} = x_{i-1,j} \qquad \text{and} \qquad \sigma_\uparrow(x)_{i,j} = x_{i,j-1}.$$

▶ **Definition 3** (Cellular automata)**.** A (two-dimensional) **cellular automaton** is a continuous action $F : \mathcal{A}^{\mathbb{Z}^2} \to \mathcal{A}^{\mathbb{Z}^2}$ that commutes with $\sigma_\rightarrow$ and $\sigma_\uparrow$. Equivalently, it can be defined by a **local rule** $\overline{F} : \mathcal{A}^{\mathbb{U}_F} \to \mathcal{A}$, where $\mathbb{U}_F \subset \mathbb{Z}^2$ is a finite **neighbourhood**, in the sense that

$$\forall x \in \mathcal{A}^{\mathbb{Z}^2}, i, j \in \mathbb{Z}^2, F(x)_{i,j} = \overline{F}((x_{(i,j)+u})_{u \in \mathbb{U}_F}).$$

This equivalence is known as the Curtis-Hedlund-Lyndon theorem [7].

2.2 Probability measures

▶ **Definition 4** (Probability measures on $\mathcal{A}^{\mathbb{Z}^2}$)**.** Let $\mathfrak{B}$ be the Borel sigma-algebra of $\mathcal{A}^{\mathbb{Z}^2}$. Denote by $\mathcal{M}(\mathcal{A}^{\mathbb{Z}^2})$ the set of probability measures on $\mathcal{A}^{\mathbb{Z}^2}$ defined on the sigma-algebra $\mathfrak{B}$. Let $\mathcal{M}_\sigma(\mathcal{A}^{\mathbb{Z}^2})$ be the **$\sigma_\uparrow, \sigma_\rightarrow$-invariant probability measures** on $\mathcal{A}^{\mathbb{Z}^2}$, that is to say the measures $\mu \in \mathcal{M}(\mathcal{A}^{\mathbb{Z}^2})$ such that $\mu(\sigma_\uparrow^{-1}(B)) = \mu(\sigma_\rightarrow^{-1}(B)) = \mu(B)$ for all $B \in \mathfrak{B}$. For a continuous application $F : \mathcal{A}^{\mathbb{Z}^2} \to \mathcal{A}^{\mathbb{Z}^2}$, denote $F\mu$ the image of the measure μ by F: $F\mu(X) = \mu(F^{-1}(X))$.

▶ **Definition 5** (Bernoulli measure)**.** The Bernoulli measure $\mu_\lambda \in \mathcal{M}_\sigma(\mathcal{A}^{\mathbb{Z}^2})$ associated with a vector $\lambda = (\lambda_a) \in [0;1]^{\mathcal{A}}$ such that $\sum_{a \in \mathcal{A}} \lambda_a = 1$ is defined as:

$$\forall u \in \mathcal{A}^{\mathbb{U}}, \mu_\lambda([u]) = \prod_{(i,j) \in \mathbb{U}} \lambda_{u_{i,j}}.$$

▶ **Definition 6** (μ-limit set)**.** Let $F : \mathcal{A}^{\mathbb{Z}^2} \to \mathcal{A}^{\mathbb{Z}^2}$ be a CA and μ an initial probability measure. The μ-limit set of F $L_\mu(F)$ is defined by:

$$u \in L_\mu(F) \iff F^t\mu([u]) \underset{t \to \infty}{\not\to} 0.$$

2.3 Compatibility

The standard Turing machine model has access to a one-dimensional working tape than can be infinite on one or both sides. We consider in this paper that the machines have access to a two-dimensional tape infinite in all directions, in order to simplify some constructions. The only difference is that the computing head, when reading the current state and the letter on the tape at its current location, has the ability to move in four different directions: $\uparrow, \downarrow, \rightarrow, \leftarrow$. This model remains exactly as powerful as a Turing machine.

▶ **Definition 7** (Computable sequence of patterns)**.** A sequence of patterns $(u_n)_{n\in\mathbb{N}} \in (\mathcal{A}^*)^{\mathbb{N}}$ is computable if there exists a Turing machine that, given as input an integer n written in binary, stops and outputs u_n.

In the previous definition, the Turing machine's alphabet contains at least $\mathcal{A}$ and $\{0, 1\}$. We can assume the input is written left to right on row 0 surrounded by a special blank state.

▶ **Proposition 8.** *Let $F : \mathcal{A}^{\mathbb{Z}^2} \to \mathcal{A}^{\mathbb{Z}^2}$ be a CA and $\mu \in \mathcal{M}_\sigma(\mathcal{A}^{\mathbb{Z}^2})$ be the uniform Bernoulli measure. Then there is a computable sequence of square patterns $(w_i)_{i\in\mathbb{N}}$ such that*

$$u \in L_\mu(F) \iff \mathit{Freq}(u, w_i) \underset{i\to\infty}{\nrightarrow} 0.$$

The sequence is built using de Bruijn tori, a combinatorial object constructed explicitly in [8]. Due to space constraints, the proof is in the appendix.

3 Main theorem

▶ **Theorem 9.** *Let μ be the uniform Bernoulli measure over $\mathcal{A}$ and $(w_i)_{i\in\mathbb{N}}$ a computable sequence of square patterns. Then there exists an alphabet $\mathcal{B} \supseteq \mathcal{A}$ and a cellular automaton F over $\mathcal{B}$ such that:*

$$u \in L_\mu(F) \iff \mathit{Freq}(u, w_i) \underset{i\to\infty}{\nrightarrow} 0.$$

This theorem along with Proposition 8 characterises all μ-limit sets when μ the uniform Bernoulli measure. The proof of the theorem relies on an explicit construction; that is, we prove the result effectively by describing the CA.

Similarly to what was done for one-dimensional CA in [2, 5], the idea is, starting from some random configuration according to a measure μ, to build a partition of connected subsets of the plane using auxiliary states. In each subset, independently, each w_i is computed successively and concatenated copies of it are written over all the subset. To ensure the density of auxiliary states tends to 0, they merge progressively in a controlled manner, offering more space for computation.

4 Construction

4.1 Overview

First, we present a sketch of the different steps of the construction corresponding to a computable sequence of patterns $(w_i)_{i\in\mathbb{N}}$. The alphabet $\mathcal{B}$ is the product of different layers, each layer being used for a different auxiliary process, plus two special states (seed and heart). The **main layer** is the writing layer whose alphabet is $\mathcal{A}$; each other layer uses a different alphabet containing a blank symbol # corresponding to the absence of information. Hence we have $\mathcal{A} \subset \mathcal{B}$ up to the bijection $a \leftrightarrow (a, \#, \ldots, \#)$.

- Colonising the space: Section 4.2.
  Starting from a random configuration drawn according to μ, we first want to "clean" the randomly generated content of the auxiliary layers. $\mathcal{B}$ contains a **seed** state $\boxed{*}$. Each seed, at time 1, erases the contents of a small area around it and give birth to membranes growing in every direction except when they meet other membranes. They erase all information contained in the auxiliary layers and membranes faking life which are recognised with the help of age counters.
- Internal metabolism: partitioning the cleaned space. Section 4.3.1.
  Each seed gives birth to a heart $\boxed{\heartsuit}$ that will be the core of a living organism. Every organism owns an age counter making sure they are all synchronised. Regularly, the organism around each living heart grows in each direction until it meets a fellow organism, thus claiming its territory.
- Internal metabolism: fighting for survival. Section 4.3.2.
  Organisms need to become larger and larger through time, so we regularly remove some of the hearts. When two hearts are too close, one of them is removed to ensure that the distance between hearts is large and tends to infinity.
- Internal metabolism: Computing and writing. Sections 4.3.4 and 4.3.5.
  In each organism, when the territory is established, some word w_n is computed and then written all over the territory. Copies of w_n thus cover the cleaned surface.

Throughout this article, t refers to the number of steps since time 0.

4.2 Colonisation of the space

4.2.1 Growing squares

There is a particular **seed** state $\boxed{*}$ that is only present in the initial configuration. It is the only relevant information in the initial configuration. Every occurrence of $\boxed{*}$ triggers the birth at time 1 and subsequent growth of a living square-shaped membrane (initially forming a 5×5 cells square).

If seeds are too close from each other and do not have enough space to form the initial organism, the northernmost seed is destroyed (westernmost in case of a tie).

A layer of the alphabet, called **cleaning layer** is dedicated to the membrane growth and cleaning process. The membrane spreads slowly to the outside, thanks to a respiration process that "pushes" the membrane to the outside. A membrane is a boundary between its inside and the outside, thus defining the direction in which it expands. To each point of the membrane is associated a binary counter that keeps track of its age (see Figure 1).

▶ **Definition 10** (Redundant binary basis)**.** Let $c = c_{n-1} \dots c_0 \in \{0, 1, 2\}^n$ be a counter. The **value** of c is $\sum_{i=0}^{n-1} c_i 2^i$ (reverse order). Since $2 = 10$, 2 can be seen as a 0 with a carry.

At each step, the counters are incremented by adding one to the least significant bit and the carries are propagated along the counter, which can be done in a local manner ($02 \to 10, 12 \to 20$).

If the membrane has sides of length n, there are n such counters on each side with the same value, with superpositions of two of them in the cells near the corner. As they grow, they need more than one cell and form a band of growing width along the membrane as shown in Figure 1. For a living membrane, the counters are created with value 0 at step $t = 1$, ensuring their age is the current time minus 1. In the other cases, the membrane and counters already existed at time $t = 0$ (with value at least 0), which means they appear older than living membranes.

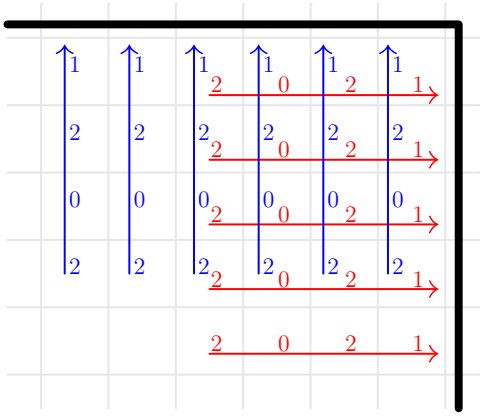

Figure 1 Corner of a membrane extending to the north and the east.

This counter is used to control the speed of the membrane. The respiration consists in taking a step forward (according to the direction of the membrane) each time the age of the counter is the exact square of an integer. The successive squares are computed under the counter, on the computation layer, using a space $O(\log t)$ if t is the age of the membrane.

We can define three kinds of membranes:

Living membranes which were created by a seed, and whose counters all have value $t-1$;

Dead membranes which have some incoherence (not closed, different counter values, no square computation...) and self-destruct when realising it;

Zombie membranes which are perfectly coherent despite not being created by a seed, and whose counters all have the same value $t' > t-1$.

The content of any cell outside a membrane is deleted, except for the encounter of another membrane. In this case the comparison process starts. The reason membranes spread slowly is to limit the interferences between the growing and comparison processes.

4.2.2 Comparison

When two membranes meet, membranes fight for survival, which is only granted to the youngest. Indeed, we saw that only living membranes can have age $t-1$, all other membranes' counters having value greater than t. Comparing the age of both counters is achieved on a dedicated **comparison layer**.

When membranes meet, two special states ⊠ are written on the comparison layer to trigger the process on each side. Each of them progresses along its corresponding counter, copying the value of the counter on the comparison layer. Incrementation and carry propagation continue in the original counter. However, it is not necessary to increment and propagate carries in the copied counter since both sides would increase by the same amount during the comparison anyway. During the copy into the comparison layer, all carries are taken into account and resolved, so that we have two pure binary counters at the end of the process.

Both copied counters progress towards the encounter point at speed 1 and a comparison is performed bit-by-bit, starting from the least significant. When the last bits of the counters arrive, we can decide which counter corresponds to the youngest membrane.

As shown in Figure 2, if at time t_1 two membranes meet, comparison of the age of counters takes place at each contact cell. Here the same process takes place at cells A, B, C and D.

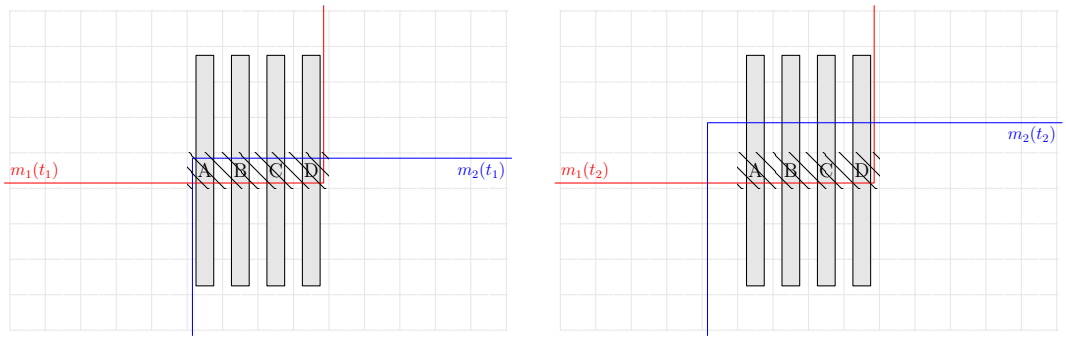

Figure 2 At time t_1, the membranes m_1 and m_2 meet on cells A, B, C and D. The counters are represented by grey areas. At t_2, when the comparison is finished, one of the squares may have grown (here m_2).

▶ **Proposition 11.** *During a comparison process, a living membrane may grow only once (including the initial growth that triggered the comparison)*

Proof. If the comparison process started at time t_0, the counters of a living membrane have length less than $\log(t_0)$. The comparison process takes at most twice as many steps as the length of the counter. The respiration process happens when t is a perfect square. Therefore the time between two successive growths, at time t_0 or later, is at least $\lceil \sqrt{t_0} \rceil$ steps. ◀

Let us consider the various possible results:

The membranes have the same age: they are both alive or both zombie. In any case, both membranes turn into a single one as shown in Figure 3. Some $\boxed{\&}$ symbols are written at the corners, so that, when both sides grow again, they remember they are part of the same membrane.

A membrane is younger: the oldest one is zombie and can be safely destroyed. A death signal $\boxed{☠}$ spread in both directions along the oldest membrane, erasing it. The surviving membrane resumes its growth, with its age counters still accurate. The same happens if a membrane grows twice, disrupting the comparison process.

Notice that only the membrane and not the "insides" of the zombie are cleaned since it can contain other living membranes. None of the signals or processes described in the following sections can enter or leave a membrane, or interact with it or counters, except if explicitly mentioned.

For $t \in \mathbb{N}$, denote

$$Pr(t) = \{F^t(c) \mid \ c \in \mathcal{B}^{\mathbb{Z}^2},\ \exists (i,j) \in \mathbb{Z}^2,\ d_\infty((i,j),(0,0)) \leq \lfloor \sqrt{t} \rfloor,\ c_{ij} = \boxed{*}\}$$

the set of images of configurations containing a seed $\boxed{*}$ at distance $\lfloor \sqrt{t} \rfloor$ at most of $(0,0)$. As μ is the uniform Bernoulli measure, the following lemma is clear:

▶ **Lemma 12.** $F^t\mu(Pr(t)) = 1 - (1 - \mu(\boxed{*}))^{(2\sqrt{t}+1)^2} \to_t 1$

This means that, with probability 1, for almost any configuration the central cell eventually belongs to the insides of a living membrane.

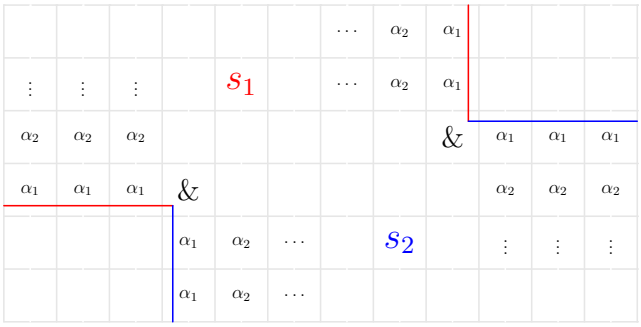

Figure 3 At the end of the comparison, if membrane counters share a common value, the common part of their boundaries is erased and [&] symbols mark the corners.

4.3 Working in the clean surface

We now consider only the protected area, which is the union of all insides of living membranes. Thus every construction presented in this section remains inside this area and stops if it reaches the membrane. They take place on four new layers: the **age**, **partitioning**, **computing** and **writing** layers.

At some time $t_n = K^{2n}, n \in \mathbb{N}$ for some integer K that will be specified later, various operations are performed simultaneously inside all membranes. First, a simulated Turing machine computes w_n. Then, repeated copies of w_n are copied everywhere inside the membrane. Meanwhile, the heart checks that it is not too close to a neighbour, and one of them is deleted if it is the case.

These operations occurs between times t_n and $t_{n+1} - 1$, which is called the n^{th} *generation*.

4.3.1 Claiming its territory

At time 1, while creating a membrane, each seed [*] transforms itself into a **heart** [♥]. Any heart is the centre of an **organism** to which it provides life. At the same time, a binary counter is given to each heart, thus giving it the knowledge of its age. This age is exactly the same for any heart inside a living membrane. This counter is the only thing contained in the age layer.

In the rest of this section, only the partitioning layer is concerned.

At time t_n, every heart send signals at speed 1 in each direction until they meet a fellow signal, in which case they disappear and the symbol [#] is written where they met. These signals erase everything on the partitioning and computing layers but disappear if they reach a membrane. In this case, [#] is written along the membrane. The **territory** of the heart $H \in \mathbb{Z}^2$ is the largest set of 4-connected cells containing H that does not contain the symbol [#]. An organism is composed of a heart and its territory.

Simultaneously, at $t = t_n$, signals leave H and draw the **body** of H: a square of size $2n + 1$ centred in H. The body is supposed to be entirely in the territory of H; if not, the organism is in conflict with every other organism whose body intersects its own. At the end of each generation, we make sure there does not remain any conflict by removing some of the hearts.

Thus, the global dynamics partition the protected space by redefining territories during each generation, then resolve conflicts: during the n^{th} generation, the distance between two surviving hearts is at least $2n - 1$ (remember we use the distance d_∞).

4.3.2 Choosing its destiny

In this section, we describe conflicts. To get organisms larger and larger through time, we want them to contain at least their entire body, whose size depends of the current generation. We need as well to control the growth of the organisms, preventing them from being too large. Indeed, we have to write the computed pattern all over the organism before the beginning of the next generation. Thus, if at some step a chain of conflicts between organisms appears, we do not want to erase all hearts simultaneously.

To avoid this, we add an algorithmic device and give to each heart some bit of information with the constraint that these bits have to be mutually independent at any given time. Then, for each conflict between two organisms, we choose the one to delete thanks to the sum of their two random bits.

First, we use two versions of the state $\boxed{*}$ in the initial configuration: $\boxed{*}_0$ and $\boxed{*}_1$. This bit is transmitted to $\boxed{\heartsuit}$ which has two versions $\boxed{\heartsuit}_0$ or $\boxed{\heartsuit}_1$. In both cases, we keep the notations $\boxed{*}$ and $\boxed{\heartsuit}$ when the value of the bit does not matter. The bit is also known by the whole boundary of the corresponding organism.

Second, note that, given some heart H living at generation n, the conflicting hearts are at distance $2n-1$ or $2n$ or they would have conflicted before. Thus, they all belong to a square of side $4n+1$ centred in H. The distance between each other is also $2n-1$ or $2n$, hence there are at most 8 simultaneous conflicts, one at most in each eighth part of the plane centred in H: NNE, ENE, ESE, SSE, SSW, WSW, WNW and NNW.

To ensure that the independence property remains true, a heart provides some fresh information to its killer when it is deleted. Hence, we give 8 other binary bits to each seed, and therefore to each heart. Each eighth part of the territory's boundary carries one of these reserve bits alongside with the main one.

During the n^{th} generation, when two organisms O and O' of hearts respectively $\boxed{\heartsuit}_b$ at (x,y) and $\boxed{\heartsuit}_{b'}$ at (x',y') meet, the sum $\beta = b \oplus b'$ is computed where the boundaries meet. If $\beta = 0$ then the northernmost heart wins (westernmost in case of a tie) and the other way around if $\beta = 1$. Then the boundary of the killed organism (say O') transmits its reserve bit b_r to the winner whose main bit becomes $b \oplus b_r$. If some organism kills many others simultaneously (at most 8), it sums all transmitted reserve bits to its own. The key point is that all main bits are and remain independent. This is ensured since the reserve bits are not used until they pass to the winner.

On the other hand, a death signal is sent to the heart of the loser, which dies at the reception. This does not interrupt the processes of computation or copying that will be described later, but the organism will never grow again and signals from other hearts will erase it during the next generation.

▶ **Definition 13.** Define the *radius* r of an organism as the largest distance from a cell inside its territory to its heart. The territory of the organism is hence bounded by $4r^2$.

▶ **Lemma 14.** *There exists a constant K, such that $p_n \to_n 1$, where p_n is the probability that at least one living heart remains in a square of radius K^n during the n^{th} generation.*

Proof. Denote $q_n, n \in \mathbb{N}$ the probability for a cell to be a living heart during generation n. For $n = 0$, $q_0 > 0$ is a constant given by μ. Then, during each generation $k \leq n$, a heart survives with probability at least $(1/2)^8$ ($1/2$ for each conflict). Hence $q_n \geq q_0 * (1/2)^{8n}$.

Two different cells have each independently probability q_n to be a heart as long as there is no chain of conflicts between them. At generation n, they have been affected only by hearts at distance $\sum_{k=0}^{n} k \leq n^2$ at most. So there are $d_n = \lfloor (2K^n+1)/(2n^2+1) \rfloor^2$ independent cells in a square of radius K^n.

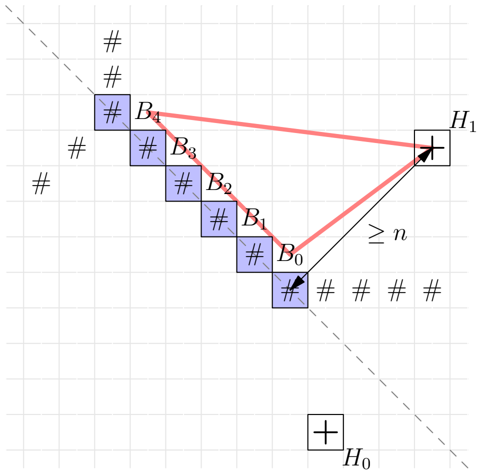

Figure 4 Two hearts H_1 and H_2 are conflicting. Cells A_0 to A_4 form the common boundary of their territory. The red triangle is a set of cells inside the territory of H_1.

Now we have $1 - p_n \leq (1 - q_n)^{d_n}$. This tends to 0 for $K \geq 17$. ◀

This lemma means that we only need to consider organisms of radius less than K^n. The other ones are sufficiently sparse.

▶ **Definition 15.** An organism is said to be *healthy* during the n^{th} generation when its radius is less than K^n (K being given in the previous lemma).

4.3.3 Shape of organisms

▶ **Lemma 16.** *If a cell A is in the organism of heart H, then each cell B such that $d_\infty(B, H) \leq d_\infty(A, H) - d_\infty(A, B)$ is in the same organism.*

Proof. The triangle inequality gives the result automatically, for any other heart H':

$$d_\infty(B, H) \leq d_\infty(A, H) - d_\infty(A, B) \leq d_\infty(A, H') - d_\infty(A, B) \leq d_\infty(B, H').$$ ◀

▶ **Lemma 17.** $F^{t_n}\mu([\boxed{\#}] \cap Pr(t_n)) = O(1/n)$

Proof. Given $n \in \mathbb{N}$, consider the set of cells containing state $\boxed{\#}$ at time t_{n+1} within the protected area. It is possible to cut this set into horizontal, vertical or diagonal segments such that each one of them is the common boundary of two specific hearts. When two hearts claim their territory, they send signals in every direction at speed one. These signals may eventually cross to give birth to the boundary. Except if they cross exactly in their corners (four cells for each organism, which is negligible), the length of their common boundary is at least 2. Consider one of these boundary segments containing cells $\{A_0, A_1, \ldots A_k\}$ and denote H_0 and H_1 the associated hearts.

The proof is illustrated on Figure 4 in the case of a diagonal segment. Denote d the line supporting the segment, as $d_\infty(H_0, H_1) \geq 2n$, $\exists j \in \{0, 1\}$ such that $d_\infty(H_j, d) \geq n$. Denote O_j the organism centred in H_j. Since $A_0, A_1, \ldots A_k$ are on the boundary of O_j, there exist distinct points $B_0, B_1, \ldots, B_{k-1}$ adjacent to $A_0, A_1, \ldots A_k$ and inside O_j.

▶ **Claim 18.** *Every cell inside the triangle $B_0 B_{k-1} H_j$ is inside O_j.*

Proof. For any such cell x, there exists $l \in [0, k-1]$ such that $d_\infty(H_j, B_l) = d_\infty(H_j, x) + d_\infty(x, B_l)$. Hence, using Lemma 16, x belongs to O_j. ◀

There are $\lfloor (k-1)(n-1)/2 \rfloor$ cells in the triangle $B_0 B_{k-1} H_j$, which means that for each cell of the boundary segment, we produced $O(n)$ cells inside an organism.

Any cell inside an organism can be attached this way to two segments at most (the border of the triangle can be shared). Thus, for any cell containing $\boxed{\#}$, there are at least $\Theta(n)$ cells that do not contain $\boxed{\#}$. Hence $F^{t_n}\mu([\boxed{\#}] \cap Pr(t_n)) = O(1/n)$. ◀

4.3.4 Computing

In this section, we deal only with the computing layer. At time $t_n, n \in \mathbb{N}$, the same computation starts around each heart. While signals leave the heart to determine the boundaries of their territory, other signals draw the limits of a square of side $\sqrt{n}$ whose down-left corner is the heart. This is the space allowed for computation. The heart creates a Turing machine head and the computation starts. It has to remain in this space and halt in less than K^{2n}.

Without loss of generality, we can choose a computable sequence of patterns $(w_i)_{i \in \mathbb{N}}$ such that w_n is the pattern computed during the n^{th} generation. Indeed, we can transform the original sequence by repeating each pattern until there is enough space and time to compute the following one. Denote $\mathbb{U}_n$ the support of w_n and l_n its size: $\mathbb{U}_n = \mathrm{supp}(w_n) = [0, l_n] \times [0, l_n]$. Considering the space allowed for computation, we have that $l_n \leq \sqrt{n}$.

4.3.5 Copying

Finally, we consider the copying layer. After computing a pattern on the computing layer of an organism, we write copies of it over the whole territory of this organism.

During the n^{th} generation, the computation takes less than K^{2n} steps, which leaves $K^{2n+2} - K^{2n}$ steps before t_{n+1}. We show that this is enough to write periodic copies of the result all over the organism, as long as the organism is healthy.

Consider an organism of heart $H = (x_H, y_H)$ during generation n. We first write 4 copies of w_n around H at $(x_H - l_n, y_H - l_n) + \mathbb{U}_n$, $(x_H - l_n, y_H) + \mathbb{U}_n$, $(x_H, y_H - l_n) + \mathbb{U}_n$ and $H + \mathbb{U}_n$. To copy a square, a machine copies all the states sequentially. First, the sides of the squares are marked on the copying layer with a state $\boxed{G}$ (this takes $O(l_n)$ steps using counters initialised with value l_n), then the machine needs $2l_n$ steps to go to the copy emplacement, make the copy and come back. There are ${l_n}^2$ cells to copy, so the whole process of copying a square takes $O({l_n}^3)$ steps.

Starting with these 4 copies of w_n, 4 different copying processes take place, each one in its quarter of the plane: north-east, north-west, south-west and south-east. We only detail the process in the north-east quarter.

The base square is copied along the vertical and horizontal axes until it reaches the limit of the territory. Simultaneously, each of these copies replicates itself in diagonal towards the north-east. This way, the whole territory is eventually covered with copies of the computed pattern w_n. The set of states $\boxed{G}$ draw a grid of step l_n. The copying process is actually a wave starting at the heart of the organism and extending the area where the pattern w_n is written. See Figure 5.

▶ **Lemma 19.** *For any healthy organism, copying takes less than $O(nK^n)$ time steps during the n^{th} generation.*

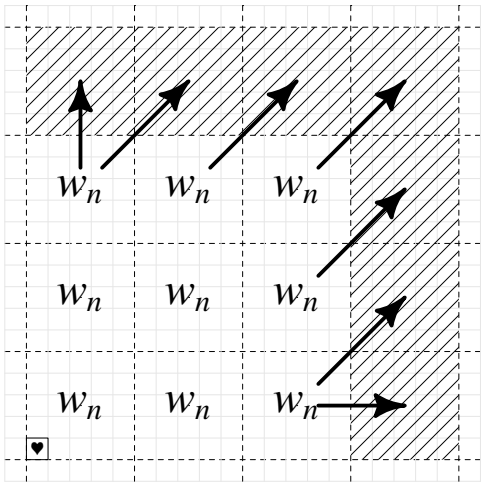

Figure 5 The square pattern is copied all over the whole territory both on axes and along diagonals, starting from the heart.

Proof. Consider a healthy organism, as the radius is bounded by K^n and the grid step is l_n, there are sequences of at most K_n/l_n square copies to do in each quarter. Each one of these copies requires $O(l_n^3)$ steps, hence the total copy time is $O(nK^n)$ (recall $l_n \leq \sqrt{n}$). ◀

▶ **Lemma 20.** *During the n^{th} generation, any cell in a healthy organism that was not reached by the copying process is at distance $\sqrt{n}$ or less of the boundary of the territory.*

Proof. Again, we prove it in the north-east quarter, the proof is symmetric in the other cases. Take a cell A in the territory of a healthy organism and at distance more than l_n of the boundary of the territory. A is in a square S of the $\boxed{G}$ grid (or would be by extending the grid). Thanks to the hypothesis we know that S entirely belongs to the organism. The copy process reached S, arriving from a square S' at the south, east or south-east of S depending of the position of S. Now, according to Lemma 16, S' entirely belongs to the organism.

This way, we can go recursively all the way back to the heart, and the copy process is necessarily successful at each step. ◀

5 Proof of the main theorem

We saw in previous sections that a configuration tends to contain only healthy organisms, and that computing and copying can be both achieved in less than $t_{n+1} - t_n$ time steps in a healthy organism. From this we now conclude.

Proof. Given a sequence $(w_n)_{n\in\mathbb{N}}$, we build the cellular automaton F over the alphabet $\mathcal{B}$ as described in the previous sections.

Suppose $t = t_{n+1} - 1, n \in \mathbb{N}$. First, if $s \in \mathcal{B} \setminus \mathcal{A}$, a cell can have state s if it is:

- outside the protected area, use Lemma 12;
- outside a healthy organism, use Lemma 14;
- in the border of a healthy organism, use Lemma 17;
- in the computation area of an organism, which are negligible since this area is a square of side $\sqrt{n}$ in territories that contain a square of side n;

- in the grid drawn in each territory (states $\boxed{G}$), negligible as well since the grid occupies less than $4l_n$ cells in each square of side l_n.

Therefore $L_\mu(F) \subseteq \mathcal{A}^*$.

Now, we show that we only need to consider the squares of the grid entirely included in a healthy organism. As we said before, it is enough to consider healthy organisms. Every square that is only partially inside a healthy field is located into a band of width less than $\sqrt{n}$ adjacent to the boundary of the field, hence there are at most $O(1/\sqrt{n})$ such cells thanks to Lemma 12. As we forced $i \leq \sqrt{n}$, we can effectively neglect those partial squares. In any other square, thanks to Lemma 20, we know that the copy was achieved successfully.

For all these reasons, for a square pattern u, $F^{t_n}\mu([u]) \sim_{n\to\infty} \mathrm{Freq}(u, w_n)$.

Moreover, during the n^{th} generation, while the copying process is engaged but not finished, some part of the main layer contains copies of w_n and the rest is still filled with copies of w_{n-1}. Hence, for some $0 \leq \alpha \leq 1$:

$$F^t\mu([u]) \sim_{n\to\infty} \left(\alpha F^{t_n}\mu([u]) + (1-\alpha)F^{t_n}\mu([u])\right).$$

◀

Perspectives

As for the one-dimensional case, we have a characterisation of all subshifts that are μ-limit sets of CA. Some corollaries can be derived from this result, but the main open problem is to generalise it to larger classes of measures. In dimension 1, the difference is that there is no need for a trick such as the one used in Section 4.3.2 to resolve conflicts while avoiding erasing too many hearts. As this trick only works with the uniform Bernoulli measure, hence, a better understanding of the dynamics of disappearance of the hearts should allow to generalise the result.

References

1 Alexis Ballier, Pierre Guillon, and Jarkko Kari. Limit sets of stable and unstable cellular automata. *Fundam. Inf.*, 110(1-4):45–57, January 2011.

2 Laurent Boyer, Martin Delacourt, Victor Poupet, Mathieu Sablik, and Guillaume Theyssier. μ-limit sets of cellular automata from a computational complexity perspective. *CoRR*, abs/1309.6730, 2014.

3 Laurent Boyer, Martin Delacourt, and Mathieu Sablik. Construction of μ-limit sets. In *JAC*, pages 76–87, 2010.

4 Laurent Boyer, Victor Poupet, and Guillaume Theyssier. On the complexity of limit sets of cellular automata associated with probability measures. In *MFCS*, pages 190–201, 2006.

5 Benjamin Hellouin de Menibus and Mathieu Sablik. Characterisation of sets of limit measures after iteration of a cellular automaton on an initial measure. *CoRR*, abs/1301.1998, 2013.

6 Martin Delacourt. Rice's theorem for -limit sets of cellular automata. In *ICALP (2)*, pages 89–100, 2011.

7 Gustav A. Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Mathematical Systems Theory*, 3(4):320–375, 1969.

8 Glenn Hurlbert and Garth Isaak. On the de bruijn torus problem. *Journal of Combinatorial Theory, Series A*, 64(1):50 – 62, 1993.

9 J. Kari. Rice's theorem for the limit sets of cellular automata. *Theoretical Computer Science*, 127:229–254, 1994.

10 P. Kůrka and A. Maass. Limit Sets of Cellular Automata Associated to Probability Measures. *Journal of Statistical Physics*, 100(5-6):1031–1047, 2000.

11 Alejandro Maass. On the sofic limit sets of cellular automata. *Ergodic Theory and Dynamical Systems*, 15:663–684, 7 1995.

12 John von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA, 1966.