mu-Limit Sets of Cellular Automata from a Computational Complexity Perspective

Laurent Boyer, Martin Delacourt, Victor Poupet, Mathieu Sablik, Guillaume Theyssier

μ-Limit Sets of Cellular Automata from a Computational Complexity Perspective

L. Boyer[a], M. Delacourt[c,*], V. Poupet[d], M. Sablik[e], G. Theyssier[b]

[a] *SAMM, Université Paris 1, 90 rue de Tolbiac, 75 634 Paris Cedex 13, France*
[b] *LAMA, Université de Savoie, CNRS, 73 376 Le Bourget-du-Lac Cedex, France*
[c] *CMM, Universidad de Chile, CNRS, Av. Blanco Encalada 2120, Santiago, Chile*
[d] *LIRMM, Université Montpellier 2, CNRS, 161 rue Ada, 34 095 Montpellier Cedex 5, France*
[e] *LATP, Université de Provence, CNRS, 39, rue Joliot Curie, 13 453 Marseille Cedex 13, France*

Abstract

This paper is about μ-limit sets of cellular automata, *i.e.* sets of configurations made of words which have a positive probability to appear arbitrarily late in the evolution, starting from an initial μ-random configuration. More precisely, we investigate the computational complexity of these sets and of decision problems concerning them. Our main results are: first, that such a set can have a Σ_3^0-hard language, second that it can contain only α-complex configurations and third that any non-trivial property concerning these sets is at least Π_3^0-hard. We also prove various complexity upper bounds, study some restriction of these questions to particular classes of cellular automata, and study different types of (non-)convergence of the probability of appearance of a word in the evolution.

Key words: cellular automata; μ-limit sets; Rice theorem; arithmetical hierarchy.

1. Introduction

A cellular automaton (CA) is a complex system defined by a local rule which acts synchronously and uniformly on the configuration space. These simple models have a wide variety of different dynamical behaviors, in particular interesting asymptotic behaviors.

In the dynamical systems context, it is natural to study the limit set of a cellular automaton: it is defined as the set of configurations that can appear arbitrarily far in time. This set captures the longterm behavior of the CA and has been widely studied since the end of the 1980s. Given a cellular automaton, it is difficult to determine its limit set. Indeed it is undecidable to know if it contains only one configuration [Kar92] and more generally, any nontrivial

*Corresponding author (mdelacourt@dim.uchile.cl)

[1] Research partially supported by the FONDECYT Postdoctorado Proyecto 3130496 and by grant 'Agence Nationale de la Recherche ANR-09-BLAN-0164'

property of limit sets is undecidable [Kar94]. Another problem is to characterize which subshift can be obtained as limit set of a cellular automaton. This was first studied in detail by Lyman Hurd [Hur87], and significant progress have been made since [Maa95, FK07] but there is still no characterization. The notion of limit set can be refined if we consider the notion of attractor [Hur90a, Kůr03].

However, these topological notions do not correspond to the empirical point of view where the initial configuration is chosen randomly, that is to say chosen according a measure μ. That's why the notion of μ-attractor is introduced by [Hur90b]. Like it is discussed in [KM00] with a lot of examples, this notion is not satisfactory empirically and the authors introduce the notion of μ-limit set. A μ-limit set is a subshift whose forbidden patterns are exactly those, whose probabilities tend to zero as time tends to infinity. This set corresponds to the configurations which are observed when a random configuration is iterated.

As for limit sets, it is difficult to determine the μ-limit set of a given cellular automaton, indeed it is already undecidable to know if it contains only one configuration [BPT06], and as for limit sets, every nontrivial property of μ-limit sets is undecidable [Del11]. In [BDS10], it was shown that large classes of subshifts such as transitive sofic subshifts can be realized as μ-limit sets.

This paper aims at pushing techniques already used in [BDS10, Del11] to their limits in order to characterize the complexity of μ-limit sets and associated decision problems. The main contribution is to show that the complexity of μ-limit sets can be much higher than that of limit sets. This fact may seem counter-intuitive given that limit sets take into acount worst-case initial conditions whereas μ-limit sets restrict to μ-typical initial configurations, thus excluding possibly complex behaviors. However our proofs show that: first, some self-organization can be achieved from random initial configurations in order to initiate more or less arbitrarily chosen computations; second, the probabilistic conditions involved in the definition of μ-limit sets allow in fact to encode more complexity in the decision problem of whether a word is accepted in the μ-limit language or not.

This article, after a section dedicated to definitions, is organized as follows :

- in Section 3 we give the detail of a generic construction we will use many times. It is similar to the ones in [BDS10, Del11] but presented as a ready-to-use tool (see Lemma 3.7).
- in Section 4 we give bounds on the complexity of the language of the μ-limit set, which in general case is Σ_3-hard, then we show that this bound can be reached. We also give a cellular automaton whose μ-limit contain only α-complex configurations.
- in Section 5, we deal with properties of μ-limit sets. First we show that every nontrivial property is at least Π_3-hard. Then we investigate the complexity of μ-nilpotency for different classes of CA.
- in Section 6 we discuss convergence issues. In particular the type of convergence: general limsup, Cesaro mean limit, simple convergence. We also show evidence of some late (non-recursive) convergence phenomena.

In the recent work [dMS13], similar constructions (with fairly different implementation details) are used, mainly to prove reachability results concerning limit probability measures obtained by iterating a CA from simple initial measures. Among other results, the set of measures that can be obtained as a simple limit is completely caracterized, and moreover, it is proven that any set of measures following a necessary computability condition and a natural "topological" condition can be achieved as a set of limit points of a sequence of measures obtained by iteration of a CA from a simple initial measure. This gives an interesting complementary point of view to the one adopted in the present paper, the link being that the μ-limit set is the closure of the union of supports of limits points of the sequence of measures obtained by iterations. However, the translation of these results into the setting of μ-limit sets is somewhat artificial, and, in any case, it does not give the complexity lower bounds established in this paper.

2. Definitions

2.1. Words and Density

For a finite set Q called an *alphabet*, denote $Q^* = \bigcup_{n\in\mathbb{N}} Q^n$ the set of all finite words over Q. The *length* of $u = u_0u_1 \dots u_{n-1}$ is $|u| = n$. We denote $Q^{\mathbb{Z}}$ the set of *configurations* over Q, which are mappings from $\mathbb{Z}$ to Q, and for $c \in Q^{\mathbb{Z}}$, we denote c_z the image of $z \in \mathbb{Z}$ by c. Denote σ the translation over the space of configurations : $\forall c \in Q^{\mathbb{Z}}, \forall z \in \mathbb{Z}, \sigma(c)_z = c_{z+1}$. For $u \in Q^*$ and $0 \leq i \leq j < |u|$, define the *subword* $u_{[i,j]} = u_iu_{i+1} \dots u_j$; this definition can be extended to a configuration $c \in Q^{\mathbb{Z}}$ as $c_{[i,j]} = c_ic_{i+1} \dots c_j$ for $i, j \in \mathbb{Z}$ with $i \leq j$. The *language* of a configuration $c \in Q^{\mathbb{Z}}$ is defined by

$$L(c) = \{u \in Q^* : \exists i \in \mathbb{Z} \text{ such that } u = c_{[i,i+|u|-1]}\}.$$

This notion extends naturally to any set of configuration $S \subseteq Q^{\mathbb{Z}}$ by taking the union. An important category of sets of configurations is that of *subshift*. A subshift is a set of configuration which is translation invariant and closed for the product topology on $Q^{\mathbb{Z}}$. Equivalently, they are sets defined by languages; a set $S \subseteq Q^{\mathbb{Z}}$ is a subshift if there is a language L of *forbidden words* defining S, *i.e.*

$$S = \{c : L(c) \cap L = \emptyset\}.$$

Subshifts are the core objects of symbolic dynamics [LM95]. Among the different kinds of subshifts, we will consider *effective subshifts*, *i.e.* those such that the forbidden language can be chosen recursively enumerable.

For every $u \in Q^*$ and $i \in \mathbb{Z}$, define the *cylinder* $[u]_i$ as the set of configurations containing the word u in position i that is to say $[u]_i = \{c \in Q^{\mathbb{Z}} : c_{[i,i+|u|-1]} = u\}$. If the cylinder is at the position 0, we just denote it by $[u]$.

For all $u, v \in Q^*$ define $|v|_u$ the *number of occurences* of u in v as:

$$|v|_u = \text{card}\{i \in [0, |v| - |u|] : v_{[i,i+|u|-1]} = u\}$$

For finite words $u, v \in Q^*$, if $|u| < |v|$, the density of u in v is defined as $d_v(u) = \frac{|v|_u}{|v|-|u|}$. For a configuration $c \in Q^{\mathbb{Z}}$, the *density* $d_c(v)$ of a finite word v is:

$$d_c(v) = \limsup_{n \to +\infty} \frac{|c_{[-n,n]}|_v}{2n+1-|v|}.$$

These definitions can be generalized for a set of words $W \subset Q^*$, we note $|u|_W$ and $d_c(W)$. We can give similar definitions for semi-configurations (indexed by $\mathbb{N}$) too.

2.2. Cellular Automata

Definition 2.1 (Cellular automaton)**.** A *cellular automaton (CA)* is a triple $\mathcal{A} = (Q_\mathcal{A}, r_\mathcal{A}, \delta_\mathcal{A})$ where $Q_\mathcal{A}$ is a finite set called *set of states* or *alphabet*, $r_\mathcal{A} \in \mathbb{N}$ is the *radius* of the automaton, and $\delta_\mathcal{A} : Q_\mathcal{A}^{2r_\mathcal{A}+1} \to Q_\mathcal{A}$ is the *local rule*.

The configurations of a cellular automaton are the configurations over $Q_\mathcal{A}$. A global behavior is induced and we will note $\mathcal{A}(c)$ the image of a configuration c given by: $\forall z \in \mathbb{Z}, \mathcal{A}(c)_z = \delta_\mathcal{A}(c_{z-r}, \dots, c_z, \dots, c_{z+r})$. Studying the dynamic of $\mathcal{A}$ is studying the iterations of a configuration by the map $\mathcal{A} : Q_\mathcal{A}^{\mathbb{Z}} \to Q_\mathcal{A}^{\mathbb{Z}}$.

When there is no ambiguity, we will note Q, r and δ for $Q_\mathcal{A}$, $r_\mathcal{A}$, $\delta_\mathcal{A}$.

A state $a \in Q_\mathcal{A}$ is said to be *permanent* for a CA $\mathcal{A}$ if for any $u, v \in Q_\mathcal{A}^r$, $\delta(uav) = a$.

2.3. Measures

We denote by $\mathcal{M}(Q^{\mathbb{Z}})$ the set of Borel probability measures on $Q^{\mathbb{Z}}$. By Carathéodory extension theorem, Borel probability measures are characterized by their value on cylinders. A measure is given by a function μ from cylinders to the real interval $[0, 1]$ such that $\mu(Q^{\mathbb{Z}}) = 1$ and

$$\forall u \in Q^*, \forall z \in \mathbb{Z}, \quad \mu([u]_z) = \sum_{q \in Q} \mu([uq]_z) = \sum_{q \in Q} \mu([qu]_{z-1})$$

A measure μ is *computable* if there exists some computable $f : Q^* \times \mathbb{Q} \to \mathbb{Q}$ (where Q is the set of states) with

$$\forall \varepsilon > 0, \forall u \in Q^*, \ \left|\mu([u]) - f(u, \varepsilon)\right| \leq \varepsilon$$

A measure μ is said to be *translation invariant* or *σ-invariant* if for any measurable set E we have $\mu(E) = \mu(\sigma(E))$.

Besides, μ is *σ-ergodic* if for any σ-invariant measurable set E we have $\mu(E) = 0$ or $\mu(E) = 1$. Finally, we say μ *has full support* if $\mu([u]) > 0$ for any word u.

Definition 2.2 (Uniform Bernoulli measure)**.** For an alphabet Q, the *uniform Bernoulli measure* μ on configurations over Q is defined by:

$$\forall u \in Q^*, i \in \mathbb{Z}, \mu([u]_i) = \frac{1}{|Q|^{|u|}}$$

Through this paper, in case no additional precision is given, μ will refer to the uniform Bernoulli measure.

For a CA $\mathcal{A} = (Q, r, \delta)$ and $u \in Q^*$, we denote for all $n \in \mathbb{N}$, $\mathcal{A}^n\mu([u]) = \mu\left(\mathcal{A}^{-n}([u])\right)$.

Definition 2.3 (Generic configuration)**.** A configuration c is said to be *weakly generic* for an alphabet Q and a measure μ if there exists a constant M such that, for any word $u \in Q^*$, $\frac{1}{M}\mu([u]) \leq d_c(u) \leq M\mu([u])$. If, moreover, any word has density $\mu([u])$, the configuration is said to be *generic*.

Remark 2.1. The set of generic configurations has measure 1 in $Q^{\mathbb{Z}}$. Which means that a configuration that is randomly generated according to measure μ is a generic configuration.

2.4. μ-Limit Sets

A μ-limit set is a subshift associated to a cellular automaton and a probability measure [KM00]. They are defined by their language as follows.

Definition 2.4 (Persistent set)**.** For a CA $\mathcal{A}$, define the *persistent set* $L_\mu(\mathcal{A}) \subseteq Q^*$ by: $\forall u \in Q^*$:

$$u \notin L_\mu(\mathcal{A}) \iff \lim_{n\to\infty} \mathcal{A}^n\mu([u]_0) = 0.$$

Then the *μ-limit set* of $\mathcal{A}$ is $\Lambda_\mu(\mathcal{A}) = \left\{c \in Q^{\mathbb{Z}} : L(c) \subseteq L_\mu(\mathcal{A})\right\}$.

Remark 2.2. Two μ-limit sets are therefore equal if and only if their languages are equal.

Definition 2.5 (μ-nilpotency)**.** A CA $\mathcal{A}$ is said to be μ-nilpotent if $\Lambda_\mu(\mathcal{A}) = \{a^{\mathbb{Z}}\}$ for some $a \in Q_\mathcal{A}$ or equivalently $L_\mu(\mathcal{A}) = a^*$.

The question of the μ-nilpotency of a cellular automaton is proved undecidable in [BPT06]. The problem is still undecidable with CA of radius 1 and with a permanent state.

Definition 2.6 (Set of predecessors)**.** Define the set of predecessors at time n of a finite word u for a CA $\mathcal{A}$ as $P^n_\mathcal{A}(u) = \left\{v \in Q^{|u|+2rn} : \mathcal{A}^n([v]_{-rn}) \subseteq [u]_0\right\}$.

The following lemma translates the belonging to the μ-limit set in terms of density in images of a weakly generic configuration.

Lemma 2.1. *Given a CA $\mathcal{A}$, a σ-invariant measure $\mu \in \mathcal{M}(Q^{\mathbb{Z}})$ and a finite word u, for any weakly generic configuration c:*

$$u \notin L_\mu(\mathcal{A}) \iff \lim_{n\to+\infty} d_{\mathcal{A}^n(c)}(u) = 0$$

Proof. Let M be such that, for any word $u \in Q^n$, $\frac{1}{M}\mu([u]) \leq d_c(u) \leq M\mu([u])$.

$$d_{\mathcal{A}^n(c)}(u) = d_c(P^n_\mathcal{A}(u)) = \sum_{v\in P^n_\mathcal{A}(u)} d_c(v)$$

$$\begin{aligned}
\sum_{v\in P^n_{\mathcal{A}}(u)} \frac{1}{M}\mu([v]) &\leq d_{\mathcal{A}^n(c)}(u) \leq \sum_{v\in P^n_{\mathcal{A}}(u)} M\mu([v]) \\
\frac{1}{M}\sum_{v\in P^n_{\mathcal{A}}(u)} \mu([v]) &\leq d_{\mathcal{A}^n(c)}(u) \leq M\sum_{v\in P^n_{\mathcal{A}}(u)} \mu([v]) \\
\frac{1}{M}\mu(\mathcal{A}^{-n}([u])) &\leq d_{\mathcal{A}^n(c)}(u) \leq M\mu(\mathcal{A}^{-n}([u])) \\
\frac{1}{M}\mathcal{A}^n\mu([u]) &\leq d_{\mathcal{A}^n(c)}(u) \leq M\mathcal{A}^n\mu([u])
\end{aligned}$$

This concludes the proof.

□

Example 2.1. We consider here the "max" automaton $\mathcal{A}_M$: the alphabet contains only two states 0 and 1. The radius is 1 and $\delta_{\mathcal{A}_M}(x, y, z) = \max(x, y, z)$.

The probability to have a 0 at time t is the probability to have 0^{2t+1} on the initial configuration, which tends to 0 when $t \to \infty$ for the uniform Bernoulli measure, so 0 does not appear in the μ-limit set. And finally $\Lambda_\mu(\mathcal{A}_M) = \{^\infty 1^\infty\}$.

The limit set of a cellular automaton is defined as $\Lambda(\mathcal{A}) = \bigcap_{i\in\mathbb{N}} \mathcal{A}^i(Q^{\mathbb{Z}})$, so $\Lambda(\mathcal{A}_M) = (^\infty 10^*1^\infty) \cup (^\infty 0^\infty) \cup (^\infty 10^\infty) \cup (^\infty 01^\infty)$. Actually, we can prove that this limit-set is an example of limit-set that cannot be a μ-limit set.

3. Construction Toolbox

3.1. Initialization: Counters and Segments

In this section we will describe a general technique that can be used to construct particular μ-limit sets. We want to build a "protected" area in a cone of the space-time diagram (the area between two signals moving in opposite directions) and make sure that nothing from the outside can affect the inside of the cone.

This construction will be used extensively throughout the article.

3.1.1. General Description

The idea is to use a special state $\boxed{*}$ that can only appear in the initial configuration (no transition rule produces this state). This state will produce a cone in which a construction will take place. On both sides of the cone, there will be unary counters that count the "age of the cone".

The counters act as protective walls to prevent the exterior from affecting the construction. Any information, appart from another counter, is erased. If two counters collide, they are compared and the youngest has priority (it erases the older one and what comes next). Because the construction is assumed to be generated by a state $\boxed{*}$ on the initial configuration, no counter can be younger since all other counters were already present on the initial configuration.

The only special case is when two counters of the same age collide. In this case they both disappear and a special delimiter state $\boxed{\#}$ is written.

3.1.2. The Younger, the Better

The $\boxed{*}$ state produces 4 distinct signals. Two of them move towards the left at speed 1/4 and 1/5 respectively. The other two move symmetrically to the right at speed 1/4 and 1/5.

Each couple of signals (moving in the same direction) can be seen as a unary counter where the value is mostly encoded in the distance between the two of them, this will be discussed later. As time goes by the signals move apart.

Note that signals moving in the same direction (a fast one and a slow one) are not allowed to cross. If such a collision happens, the slower signal is erased. A collision cannot happen between signals generated from a single $\boxed{*}$ state but could happen with signals that were already present on the initial configuration. Collisions between counters moving in opposite directions will be explained later as their careful handling is the key to our construction.

Because the $\boxed{*}$ state cannot appear elsewhere than on the initial configuration and counter signals can only be generated by the $\boxed{*}$ state (or be already present on the initial configuration), a counter generated by a $\boxed{*}$ state is at all times the smaller possible one: no two counter signals can be closer than those that were generated together. Using this property, we can encapsulate our construction between the smallest possible counters. We will therefore be able to protect it from external perturbations: if something that is not encapsulated between counters collides with a counter, it is erased. And when two counters collide we will give priority to the youngest one.

3.1.3. Dealing with collisions

Collisions of signals are handled in the following way:

- nothing other than an *outer* signal can go through another *outer* signal (in particular, no "naked information" not contained between counters);
- when two *outer* signals collide they move through each other and comparison signals are generated as illustrated by Figure 1:
  - on each side, a signal moves at maximal speed towards the *inner* border of the counter, bounces on it (C and C') and goes back to the point of collision (D);
  - the first signal to come back is the one from the youngest counter and it then moves back to the *outer* side of the oldest counter (E) and deletes it;
  - the comparison signal from the older counter that arrives afterwards (D') is deleted and will not delete the younger counter's *outer* border;
  - all of the comparison signals delete all information that they encounter other than the two types of borders of counters.

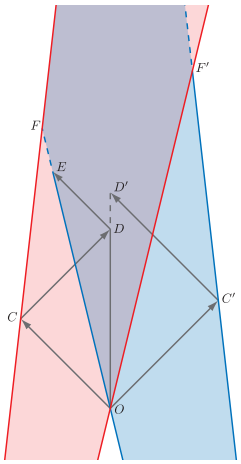

Figure 1: The bouncing signal must arrive (point E) before the older counter moves through the younger one (point F).

Counter Speeds. It is important to ensure that the older counter's *outer* border is deleted before it crosses the younger's *inner* border. This depends on the speeds s_o and s_i of the *outer* and *inner* borders. It is true whenever $s_o \geq \frac{1-s_i}{s_i+3}$. If the maximal speed is 1 (neighborhood of radius 1), it can only be satisfied if

$$s_i < \sqrt{5} - 2 \simeq 0.2360$$

This means that with a neighborhood of radius 1 the *inner* border of the counter cannot move at a speed greater than $(\sqrt{5}-2)$. Any rational value lower than this is acceptable. For simplicity reasons we will consider $1/5$ (and the corresponding $1/4$ for the *outer* border of the counter). If we use a neighborhood of radius k, the counter speeds can be increased to $k/5$ and $k/4$.

Exact Location. Note that a precise comparison of the counters is a bit more complex than what has just been described. Because we are working on a discrete space, a signal moving at a non integer speed does not actually move at each step. In particular, in the case of radius 1, it stays on one cell for a few steps before advancing, but this requires multiple states.

In such a case, the cell of the signal is not the only significant information. We also need to consider the current state of the signal: for a signal moving at speed $1/n$, each of the n states represents an advancement of $1/n$, meaning that if a signal is located on a cell c, depending on the current state we would consider it to be exactly at the position c, or $(c+1/n)$, or $(c+2/n)$, etc. By doing so we can have signals at rational non-integer positions, and hence consider that the signal really moves at each step.

When comparing counters, we will therefore have to remember both states of the faster signals that collide (this information is carried by the vertical signal) and the exact state in which the slower signal was when the maximal-speed

signal bounced on it. That way we are able to precisely compare two counters: equality occurs only when both counters are exactly synchronized.

The Almost *Impregnable Fortress.* Let us now consider a cone that was produced from a $\boxed{*}$ state on the initial configuration. As it was said earlier, no counter can be younger that the ones on each side of this cone. There might be other counters of exactly the same age, but then these were also produced from a $\boxed{*}$ state and we will consider this case later and show that it is not a problem for our construction.

Nothing can enter this cone if it is not preceded by an *outer* border of a counter. If an opposite *outer* border collides with our considered cone, comparison signals are generated. Because comparison signals erase all information but the counter borders, we know that the comparison will be performed correctly and we do not need to worry about interfering states. Since the borders of the cone are the youngest possible signals, the comparison will make them survive and the other counter will be deleted.

Note that two consecutive opposite *outer* borders, without any *inner* border in between, are not a problem. The comparison is performed in the same way. Because the comparison signals cannot distinguish between two collision points (the vertical signal from O to D in Figure 1) they will bounce on the first they encounter. This means that if two consecutive *outer* borders collide with our cone, the comparisons will be made "incorrectly" but this error will favor the well formed counter (the one that has an *outer* and an *inner* border) so it is not a problem to us.

Evil Twins. The last case we have to consider now is that of a collision between two counters of exactly the same age. Because the only counters that matters to us are those produced from the $\boxed{*}$ state, the case we have to consider is the one where two cones produced from a $\boxed{*}$ state on the initial configuration collide.

According to the rules that were descibed earlier, both colliding counters are deleted. This means that the right side of the leftmost cone and the left part of the rightmost cone are now "unprotected" and facing each other. A delimiter state $\boxed{\#}$ is then written and remains where the collision happened, as illustrated in Figure 2.

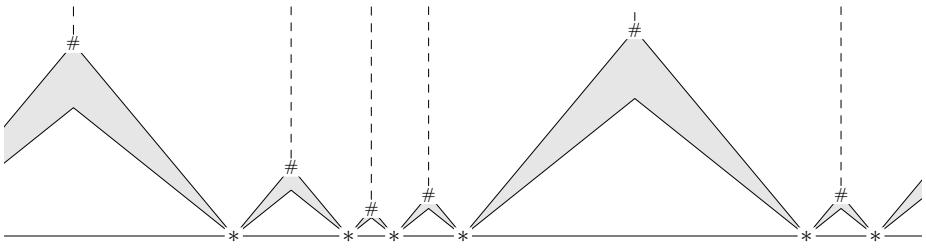

Figure 2:

A $\boxed{\#}$ state is said to be *acceptable*, if it has been written during the collision of counters created by $\boxed{*}$.

Lemma 3.1. *There exists a constant K_c such that if two acceptable $\boxed{\#}$ are distant of k, they appeared before time $k \times K_c$.*

Proof. Given an initial configuration c, consider two acceptable $\boxed{\#}$ at positions x and $x+k$ for $x \in \mathbb{N}$ which appear respectively at times $t_1 \in \mathbb{N}$ and $t_2 \in \mathbb{N}$. They were produced by the collision of counters created by $\boxed{*}$ states. Considering the speed s_1 of the outer border of a counter, there exists $\varepsilon_1, \varepsilon_2 \in \{0, 1\}$ such that $c_{x-s_1t_1} = c_{x+s_1t_1+\varepsilon_1} = c_{x+k-s_1t_2} = c_{x+k+s_1t_2+\varepsilon_2} = \boxed{*}$.

As an acceptable $\boxed{\#}$ can only be produced by signals coming from consecutive $\boxed{*}$, it is required that $x+s_1t_1+\varepsilon_1 \leq x+k-s_1t_2$. Then, we have $k \geq s_1(t_1+t_2)$. Denote $K_c = \frac{1}{s_1}$. We proved that $t_1 \leq kK_c$ (respectively $t_2 \leq kK_c$). □

Each produced $\boxed{\#}$ needs to ensure that it has neighbors on each side. To detect neighbors, it sends a signal to the left and waits for a similar signal from the right. The speed of these signals must be s_i or smaller. When a $\boxed{\#}$ receives such a signal, it becomes *activated*.

Definition 3.1 (Segment)**.** A *segment* is the set of cells between two consecutive acceptable $\boxed{\#}$.

When the left $\boxed{\#}$ of a segment becomes activated, the segment itself is said to become *activated*.

This is essential since most cells will belong to segments, as stated by the following lemma.

Lemma 3.2. *For any generic configuration c, the probability for a cell at time t to be outside activated segments tends to 0 when t grows.*

Proof. Consider a predecessor uav (with $|u| = |v|$) at time 0 of such a state. Necessarily, either u or v does not contain any subword in $\boxed{*}Q_{\mathcal{A}}^{*}\boxed{*}$. This concludes the proof.

□

We will see in the next sections what happens to the segments. The idea is to allow computation in each segment; but in order to get arbitrarily large space for computation, some delimiters will be erased to pool the available space of many segments. In broad outline, each segment will contain a computation done by one Turing machine that has access only to the space inside the segment. We will make sure that most segments eventually merge with another one, which means most segments become arbitrarily large through time.

Merging means that the $\boxed{\#}$ delimiter between two segments is erased, and hence a new larger segment replaces the previous ones. A segment will be called *successor* when it has been created by a merging, then all the segments it replaces are its *ancestors*. If a segment is not a successor, it will be called *initial*. For each merging, at some step, which depends on the merging process, the new segment becomes activated.

3.2. Computation inside segments

In the next sections of this article, we will use many times an automaton with counters, and we will add some computation inside segments. For each result we will prove with this technique, we will describe the computation that has to be run in segments. But there are general ideas and constraints that will be respected in every case. Consider an automaton that contains a copy of our cellular automaton with counters and segments, to describe the computation in a segment of length n, we require that:

- the computation and writing of the result are achieved by the head of a Turing machine launched when the segment becomes activated;
- there are two layers for the machine, one for computation, one for writing; formally this is achieved using a cartesian product of alphabet $Q_c \times Q_o$;
- the computation itself uses only $i = O(\log n)$ consecutive cells, even if these cells are not necessarily always the same through the whole computation; formally the computation alphabet Q_c contains a special *blank* state b and at most $O(\log n)$ cells of the segment have a computation state different from b.
- the Turing machine of a segment writes only once in the writing layer of each cell, from the left to the right, using the alphabet Q_o for the output of the machine;
- when some segments merge together, nothing is erased on the writing layer, it will be covered step by step by the new output; the now inside $\boxed{\#}$ are replaced by a special state $(b, \boxed{€})$, $\boxed{€}$ being a state never written by the Turing machine.

For any segment v at time t, we denote $w_t(v)$ the content of v. In what follows, couples (b, a) where $a \in Q_o$ will often be assimilated to a.

Therefore, for any v and t, $w_t(v)$ is the concatenation of two subwords: the beginning of the result of the computation in the segment, and the end of the results written by its predecessors, which may contain $\boxed{€}$ states. One of those two parts may be empty.

3.3. Synchronous Merging Process

Now, let us describe the dynamics of segments among themselves. In particular we will specify particular times when merging can happen, independently from the computation performed inside each segment. We will fix a lower bound on the acceptable size of a segment, and at these specific times, any segment that is smaller than this bound will merge. For this purpose we need to synchronize all the segments. As counters compute the time since the initial configuration, we will keep this information in segments. Therefore, time since the initial configuration is a knowledge shared by every segment. With such a protocol, mergings are many to one and not only two to one. We now describe the automaton $\mathcal{A}_s$ which behaves as explained with counters and segments.

3.3.1. Synchronization

When a $\boxed{\#}$ is created by the collision of two counters, their common value of time is written in base K, for some $K \geq 2$, on each side of the $\boxed{\#}$. Hence, the age of each segment is written on both its sides. And every such K-ary counter keeps computing time. As any segment is delimited by acceptable $\boxed{\#}$, this age is the same for all of them and is stored within $\lceil \log_K(t) \rceil$ cells on each side.

At time t, a segment will be admissible if its length n is such that $\lceil \log_K(t) \rceil \leq \lfloor \sqrt{n} \rfloor$. To test this condition, segments will measure their own length. This is achieved by sending a signal from the left delimiter to the right one and back. The signal will count the length in base K, then $\lfloor \sqrt{n} \rfloor$ is computed and written on both sides of the segment. Now each segment knows its age and its size.

Denote $t_i = K^i$ for all $i \in \mathbb{N}$. We allow segments to merge at time t_i for any $i \in \mathbb{N}$. As $\lceil \log_K(t) \rceil$ remains unchanged between $t = t_i$ and $t = t_{i+1}$ for any i, each segment has to decide before $t = t_i$ if $i + 1 \leq \lfloor \sqrt{n} \rfloor$. If not, the segment decides to merge.

3.3.2. Colors

As we have determined specific times for mergings, computation will not be ended in large segments, and this could lead to difficulties. To avoid having too many such segments, we use a trick to rarefy mergings of a lot of segments together. Each segment will be colored in B or R, and each $\boxed{\#}$ will have a bit of additional information. These bits on the delimiters will let us give colors to successor segments. Therefore, we have two delimiter states $\boxed{\#}_0$ and $\boxed{\#}_1$ that replace the unique $\boxed{\#}$. When the information of this bit is not relevant, we will still speak of $\boxed{\#}$ for $\boxed{\#}_0$ or $\boxed{\#}_1$ indistinctly. The special state $\boxed{*}$ is replaced by four states $\boxed{*}_0^B$, $\boxed{*}_0^R$, $\boxed{*}_1^B$ and $\boxed{*}_1^R$. Then, the initial segment generated by a state $\boxed{*}_i^R$ is R colored and B colored otherwise. And the bit i of some state $\boxed{*}_i^C$ is transmitted to the delimiter $\boxed{\#}_i$ that it produces on its left as shown on Figure 3. The color of a segment is remembered on both its extremities.

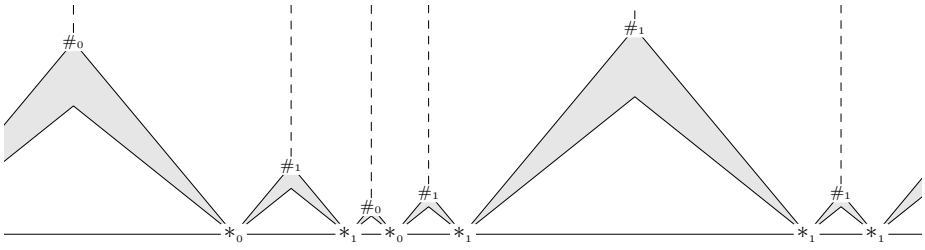

Figure 3:

3.3.3. Merging

For some $i \in \mathbb{N}$, each segment has to decide whether it will need to merge at time t_i (if it is smaller than $(i+1)^2$). If so, it checks whether its neighbors want to merge too, and what colors they wear. Then the rules to choose which neighbor it will merge with, are the following:

- if none of its neighbors wants to merge, it merges with the left one,
- if one only among its neighbors wants to merge, it merges with that one,
- if both want to merge, it merges with one having the other color if possible, and in case of indecision, with the left neighbor.

Then each $\boxed{\#}$ delimiter between a segment and the segment it wants to merge with is erased and replaced by a $\boxed{\in}$. New segments are created between the remaining $\boxed{\#}$. The color of the new segment is determined by the bit of the leftmost $\boxed{\#}$ erased inside it: if this bit was 0, then the segment is R colored, and else B. Synchronous mergings are illustrated on Figure 4.

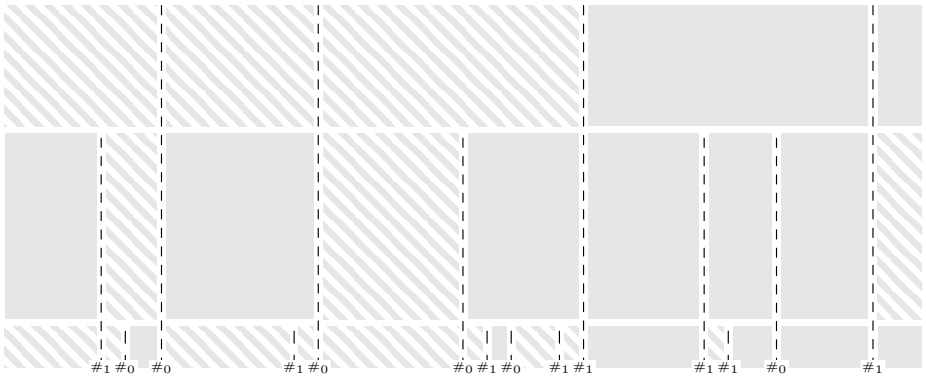

Figure 4:

Remark 3.1. To prepare itself, a segment that needs to merge before t_{i+1} (suppose we are at timestep $t = t_i$) has to:

- compute its length n, which needs $2n$ timesteps;
- compute $(i+1)^2$ which takes time polynomial in i;
- compare both, linear time;
- check its neighbors: n timesteps (if we suppose they have achieved their own computations).

A segment needs to merge if $n \leq (i+1)^2$, each of these steps requires only polynomial time in i, and for large enough i (large enough time), this is achieved in less than $(t_{i+1} - t_i)$ timesteps. So each segment that needs to merge has enough time to decide it before the merging step t_{i+1}. Other segments declare nothing to their neighbors, meaning they do not want to merge.

So mergings can concern:

- either many segments that all want to merge,
- or one that wants to merge and one that does not.

In this synchronous merging process, a segment contains the writing and computing layers, and time counters on each side of the segment.

Remark 3.2. 1. For any $i \in \mathbb{N}$, after time t_i, each segment is larger than i^2.

2. If two segments exactly merge at time $t_i, i \in \mathbb{N}$, at least one of them is smaller than i^2.

3. If three or more segments merge together at time $t_i, i \in \mathbb{N}$, they are all smaller than i^2.

4. On the images of a generic configuration by $\mathcal{A}_s$, colors' distribution of the segments remains random according to μ through time.

Proof. Distribution of $\boxed{*}$ states is random according to μ on the initial configuration, hence colors' distribution is random too over initial segments. And when segments merge, the color of the new one is chosen independently from the ancestors' or neighbors' one. Therefore, the distribution remains random. □

3.3.4. μ-limit sets

The following proposition is obvious :

Proposition 3.3. $\boxed{\#} \notin L_\mu(\mathcal{A}_s)$

Proof. Take a generic configuration c, the density α_t of cells outside segments at time t tends to 0. For any $i \in \mathbb{N}$, the size of any segment at $t \geq t_i$ is i^2 at least. Therefore, the density of $\boxed{\#}$ states is $d_{\mathcal{A}_s^t(c)}(\boxed{\#}) \leq \alpha_t + \frac{1}{i^2}$, which tends to 0. We conclude thanks to Lemma 2.1. □

In the following sections, we will use this automaton and include computations in the segment. We will need to characterize the μ-limit set of such an automaton. To simplify this study, we will prove that we can look at some particular segments only. Indeed, it is much easier to know what contain segments that have finished their computation. Hence we consider the following notion:

Definition 3.2. For all $i \in \mathbb{N}$, a segment is said to be *acceptable* at time $t_i \leq t < t_{i+1}$ if its size is less than $K^{i/4}$. For a configuration c, we will denote $S_t(c)$, the set of acceptable segments that appear in $\mathcal{A}_s^t(c)$.

Remark 3.3. As the size of acceptable segments is bounded, the set of all acceptable segments at time $t \in \mathbb{N}$ for all configurations is finite. Denote it S_t. Note that $S_t = S_t(c)$ for any generic configuration c and any full-support measure.

The following lemma shows that acceptable segments tend to cover the whole image configurations.

Lemma 3.4. *Starting from an initial generic configuration c :*

$$P_{x\in\mathbb{Z}}\left[\exists a, b \in \mathbb{Z}, a \leq x \leq b, \mathcal{A}^t(c)_{[a,b]} \in S_t\right] \xrightarrow[t\to+\infty]{} 1$$

Proof. Consider a generic configuration c.

Consider time $t_i \leq t < t_{i+1}$ for large enough i (such that $K^{i/4} - K^{(i-1)/4} \geq 2i^2$ and $K^{i/4} > 2i^2$). If a segment is larger than $K^{i/4}$ at time t, then it is necessarily:

(a) either a segment that did not merge.

(b) either the successor of two segments exactly.

(c) either an initial large segment.

(d) or a segment produced by the merging of more than three segments.

Consider a case **(b)** segment of length k, the size of at least one of the segments that merged was less than i^2. The other segment was then necessarily larger than $K^{i/4} - i^2 \geq K^{(i-1)/4}$. This means that this segment was not acceptable before t_i.

We consider the chain of non acceptable predecessors in case **(a)** or **(b)** of the segment at time K^j, $j \leq i$. The oldest segment of this chain is either case **(c)** or case **(d)**: there exists $h \leq i$ minimal such that at time K^j, for all $h \leq j \leq i$, one predecessor at least of the segment is larger than $K^{j/4}$.

This predecessor was too large, and hence did not want to merge, so it merged with at most one small segment at each t_j. If this predecessor's size at t_h was l, its size at t_i is $k \leq l + \sum_{j=0}^{i} j^2$.

There exists i_0 such that $\forall i \geq i_0$, $\sum_0^i j^2 \leq \frac{1}{2}K^{i/4}$, therefore, as $l + \sum_0^i j^2 \geq k \geq K^{i/4}$, we have $l \geq \frac{1}{2}K^{i/4}$ and $k - l \leq \frac{1}{2}K^{i/4}$. Hence, the segment's size doubled at most between steps h and i.

Therefore non acceptable segments come from case **(c)** or case **(d)** and have at most doubled.

We consider now two different cases only:

(i) initial large segments and their successors.

(ii) segments produced by the merging of many segments and their successors.

Denote d_i the probability for a cell to be in a case **(i)** non acceptable segment at time t_i, and e_i the probability for a cell to be in a case **(ii)** non acceptable segment at time t_i.

Claim 3.5. *The probability d_i for case* **(i)** *segments tends to* 0.

Proof. Denote q the number of states of $\mathcal{A}_s$. An initial segment of length m is produced by three $\boxed{*}$ distant from l_1 and l_2, with $(l_1 + l_2)/2 = m$. So its probability of apparition is less than $\frac{4}{q}\left(\frac{q-4}{q}\right)^{l_1}\frac{4}{q}\left(\frac{q-4}{q}\right)^{l_2}\frac{4}{q}$. Considering the $2m$ possibilities for the choice of l_1 and l_2, the probability is less than $2m\left(\frac{4}{q}\right)^3\left(\frac{q-4}{q}\right)^{2m}$.

As the size of such a segment doubled at most since its creation, we consider initial segments of length at least $\frac{1}{2}K^{i/4}$. And the current size is less than twice

the size of the initial segment. Therefore the density induced by those segments is:

$$\begin{aligned} d_i &\leq \sum_{m \geq \frac{1}{2}K^{i/4}} 2m \left(2m \left(\frac{4}{q}\right)^3 \left(\frac{q-4}{q}\right)^{2m} \right) \\ d_i &\leq 4\left(\frac{4}{q}\right)^3 \sum_{m \geq \frac{1}{2}K^{i/4}} m^2 \left(\frac{q-4}{q}\right)^{2m} \end{aligned}$$

And $d_i \to_{i\to\infty} 0$.

□

Claim 3.6. *The probability e_i for case* **(ii)** *segments tends to* 0.

Proof. Consider a segment of size m successor of more than 3 segments. If the merging happened at time t_h, all the predecessors of it were smaller than h^2, or they would not have merged together. So $p = \frac{m}{h^2}$ segments at least have merged. Now consider the colors of these segments. Due to the merging rules, they prefer to merge with a segment of the opposite color, and if not possible on their left, so the colors' distribution among them was:

$$R(RB)^r R^{p-2r-1},\ (RB)^r R^{p-2r},\ R(RB)^r B^{p-2r-1} \text{ or } (RB)^r B^{p-2r}$$

(or symmetrically if starting with B) for some $r \in \mathbb{N}$. So the distribution is determined by its shape (among eight possible shapes), $r \leq p/2$ and its length. Therefore the probability of such a succession of colors is less than $4\frac{m}{h^2}2^{-\frac{m}{h^2}}$.

Now we consider all case **(ii)** segments at time t_i. As precedently, their size is at most $2m$ and they were created at time t_h with a size $m \geq \frac{1}{2}K^{i/4}$. So the corresponding density is:

$$\begin{aligned} e_i &\leq \sum_{h \leq i} \sum_{m \geq \frac{1}{2}K^{i/4}} 2m \left(\frac{4m}{h^2} 2^{-\frac{m}{h^2}} \right) \\ e_i &\leq i \sum_{m \geq \frac{1}{2}K^{i/4}} 2m \left(4m \times 2^{-\frac{m}{i^2}} \right) \end{aligned}$$

And $e_i \to_{i\to\infty} 0$.

□

Finally, thanks to these two claims, we proved that the probability for a cell to be in an acceptable segment tends to 1. □

We show now that, given an automaton designed this way, its μ-limit set depends only on the content of the acceptable segments at times $(t_i - 1)_{i\in\mathbb{N}}$.

Lemma 3.7. *Given a generic configuration c and $u \in Q^*$, we have :*

- *if* $\exists \varepsilon > 0, \forall i_0 \in \mathbb{N}, \exists i \geq i_0, \forall s \in S_{t_i}, d_{w_{t_{i+1}-1}(s)}(u) \geq \varepsilon$ *then* $u \in L_\mu(\mathcal{A})$*;*
- *if* $\forall \varepsilon > 0, \exists i_0 \in \mathbb{N}, \forall i \geq i_0, \forall s \in S_{t_i}, d_{w_{t_{i+1}-1}(s)}(u) \leq \varepsilon$ *then* $u \notin L_\mu(\mathcal{A})$.

Proof. Let $u \in Q^*$.

First suppose that $\exists \varepsilon > 0, \forall i_0 \in \mathbb{N}, \exists i \geq i_0, \forall s \in S_{t_i}, d_{w_{t_{i+1}-1}(s)}(u) \geq \varepsilon$. Thanks to Lemma 3.4, there exists $i_0 \in \mathbb{N}$, such that

$$\forall i \geq i_0, \forall x \in \mathbb{Z}, P_{x \in \mathbb{Z}} \left[\exists a, b, a \leq x \leq b, \mathcal{A}^{t_i}(c)_{[a,b]} \in S_{t_i} \right] \geq 1/2$$

The hypothesis of the lemma allows us to find $i_{j+1} \geq i_j + 1, j \geq 1$ such that with $\tau_j = t_{i_j+1} - 1$, we get :

$$\forall s \in S_{t_{i_j}}, d_{w_{\tau_j}(s)}(u) \geq \varepsilon$$

Finally, $\forall j \geq 0, d_{\mathcal{A}^{\tau_j}(c)}(u) \geq \varepsilon/2$ and thanks to Lemma 2.1, we conclude that $u \in L_\mu(\mathcal{A})$.

Now if $\forall \varepsilon > 0, \exists i_0 \in \mathbb{N}, \forall i \geq i_0, \forall s \in S_{t_i}, d_{w_{t_{i+1}-1}(s)}(u) \leq \varepsilon$, take some $\varepsilon > 0$. Lemma 3.4 concerns acceptable segments, and with the same arguments, we can generalize it to acceptable segments successors only of acceptable segments. Denote Σ_t this set of acceptable segments successors of acceptable segments only at time t. For some $T \in \mathbb{N}$:

$$\forall t \geq T, P_{x \in \mathbb{Z}} \left[\exists a, b \in \mathbb{N}, a \leq x \leq b, \mathcal{A}^t(c)_{[a,b]} \in \Sigma_t \right] \geq 1 - \varepsilon$$

Due to the hypothesis, there exists $i_0 \in \mathbb{N}$, such that $t_{i_0} \geq T$ and $\forall i \geq i_0, \forall s \in S_{t_i}, d_{w_{t_{i+1}-1}(s)}(u) < \varepsilon$. For a segment s of length k at time $t_{i+1} \geq t \geq t_i, i \geq i_0$, we can write $w_t(s) = v_1 v_2$ where, except for the values of time and length :

- v_1 is a prefix of $w_{t_{i+1}-1}(s)$;
- v_2 is a suffix of $w_{t_i}(s)$, that is a concatenation of words written in acceptable segments at time $t_i - 1$.

In the word v_2 at time t, due to the construction, there can be some states $\boxed{€}$ that replaced the $\boxed{\#}$ erased during the merging step at t_i. In any case, the density of u in s is lower than the density of u in acceptable segments at time $t_j - 1, j \geq i_0$.

Therefore, $d_{\mathcal{A}^t(c)}(u) \leq (1-d) + d.d_1$ where :

- $d \geq 1 - \varepsilon$ is the probability for a cell to be in a segment of Σ;
- $d_1 \leq \varepsilon$ is the density of u in acceptable segments.

Finally $d_{\mathcal{A}^t(c)}(u) \to_t 0$.

□

Corollary 3.8. *For any word u in $Q^* \backslash Q_o^*$, $u \notin L_\mu(\mathcal{A})$.*

4. Building complex μ-limit sets

4.1. Complexity upper-bounds

Before giving examples of complex μ-limit sets, let's establish some upper bounds.

A word w is a *wall* for a CA F if for any $c, c' \in [w]_0$ we have:

1. if $c_z = c'_z$ for any $z < 0$ then $F^t(c)_z = F^t(c')_z$ for any $z < 0$ and any $t \geq 1$
2. if $c_z = c'_z$ for any $z \geq |w|$ then $F^t(c)_z = F^t(c')_z$ for any $z \geq |w|$ and any $t \geq 1$

It is well-known that a one-dimensional CA F has equicontinuous points if and only if it has walls [Kůr97].

The following proposition is a generalization of theorem 1 of [BPT06] to a broader class of measures.

Proposition 4.1. *Let μ be a σ-ergodic measure with full support and F a CA admitting w as a wall. Then $L_\mu(F)$ is exactly the set of words occuring in the (temporal) period of the orbit of some (spatially) periodic configuration of period wu for some u, formally:*

$$v \in L_\mu(F) \iff \exists t, p \geq 1, v_1, v_2, u \text{ such that } \begin{cases} F^t\big({}^\omega(wu)^\omega\big) = {}^\omega(v_1vv_2)^\omega \text{ and,} \\ F^p\big({}^\omega(v_1vv_2)^\omega\big) = {}^\omega(v_1vv_2)^\omega \end{cases}$$

Proof. First, consider some word v occuring in the period of the orbit of ${}^\omega(wu)^\omega$ as in the proposition. Then, for each $k \geq 0$, we have $[wuw] \subseteq F^{-t-kp}\big([v_1vv_2]\big)$ because w is a wall for F. Hence $F^{t+kp}\mu([v]) \geq \mu([wuw]) > 0$ because μ has full support, which shows $v \in L_\mu(F)$.

Suppose now that $v \in L_\mu(F)$. By definition there is $\varepsilon > 0$ and a sequence (t_n) such that, for all n, $F^{t_n}\mu([v]) \geq \varepsilon$. Consider for any $k \geq 0$ the set:

$$X_k = \bigcup_{-k \leq i \leq k} [w]_i$$

The union $X = \cup_{k \geq 0} X_k$ has measure 1 because μ is σ-ergodic, X is σ-invariant, $[w]_0 \subseteq X$ and μ has full support. Moreover the sequence X_k is increasing, so there is k_0 such that $\mu(X_{k_0}) > 1 - \frac{\varepsilon}{2}$. By σ-invariance of μ we deduce that the set

$$Y = \sigma^{k_0+|w|}(X_{k_0}) \cap \sigma^{-k_0-|v|-1}(X_{k_0})$$

is such that $\mu(Y) > 1 - \varepsilon$. Hence, for any n, $F^{-t_n}([v]) \cap Y \neq \emptyset$. We deduce that there is some sub-sequence (t_{n_p}) such that, for some $i < |w|$ and $j > |v|$, and for any p, $F^{-t_{n_p}}([v]) \cap [w]_i \cap [w]_j \neq \emptyset$ (recall that σ is the "left" shift). Using the fact that w is a wall, we conclude that v occurs in the (temporal) period of the orbit of some (spatially) periodic configuration of period wu for some u. □

Theorem 4.2. *Let $\mathcal{A}$ be any CA and μ a translation invariant measure. We have the following upper bounds:*

- *if μ is computable then $L_\mu(\mathcal{A})$ is a Σ_3^0 arithmetical set;*
- *if μ is σ-ergodic with full support and $\mathcal{A}$ has equicontinuity points, then $L_\mu(\mathcal{A})$ is recursively enumerable.*

Proof. Since μ is computable by some function $f : A^* \times \mathbb{Q} \to \mathbb{Q}$, there is a computable function $g : A^* \times \mathbb{Q} \times \mathbb{N} \to \mathbb{Q}$ such that for any ε, any $t \in \mathbb{N}$ and any u:

$$\left|\mathcal{A}^t\mu([u]_0) - g(u, \varepsilon, t)\right| \leq \varepsilon.$$

Indeed, it is sufficient to compute $\mathcal{A}^{-t}(u)$ and sum $f(v, \varepsilon')$ for all elements v of this set and a computably small enough ε'. Then, from the definition of $L_\mu(\mathcal{A})$ we have

$$u \notin L_\mu(\mathcal{A}) \Leftrightarrow \forall \varepsilon > 0, \exists t_0, \forall t \geq t_0, g(u, \varepsilon, t) \leq \varepsilon.$$

Therefore $L_\mu(\mathcal{A})$ is Σ_3^0.

Now suppose that μ is σ-ergodic with full support and that $\mathcal{A}$ has equicontinuous points. By hypothesis $\mathcal{A}$ admits some wall w (see [Kůr97]). Therefore Proposition 4.1 ensures that $L_\mu(\mathcal{A})$ is the set of words occuring in the (temporal) period of the orbit of some (spatially) periodic configuration of period wu for some u. Since, the temporal cycle reached from a spatially periodic initial configuration is finite and recursively bounded in the size of the spatial period, $L_\mu(\mathcal{A})$ is recursively enumerable. □

4.2. Σ_3-hard example

Here we will prove that the μ-limit language of a cellular automaton can have complexity Σ_3-hard. For that, with the help of the construction described in Section 3, we will prove a reduction to a Σ_3-hard problem on Turing machines.

Definition 4.1. A Turing machine M is said to be *co-finite* (and we write $M \in COF$) when there exists $i_0 \in \mathbb{N}$ such that M halts on every input $i \geq i_0$.

The following result was proved in [Odi99].

Theorem 4.3. *The problem COF has complexity Σ_3-hard.*

Now we can prove that :

Theorem 4.4. *There exists a cellular automaton $\mathcal{A}$ such that $L_\mu(\mathcal{A})$ is Σ_3-complete.*

Proof. We already know that this problem is Σ_3 at most. We will use Lemma 3.7 to prove this theorem. Hence, the automaton will have the structure described in Section 3 and we only have to consider the words $w_{t_i}(s)$ for $i \in \mathbb{N}$ and acceptable segments s.

Let us describe the computation inside segments. First consider a computable enumeration f of $\mathbb{N}^3$, such that for any $(j, k, l) \in \mathbb{N}^3$ there exist infinitely many $i \in \mathbb{N}$ with $f(i) = (j, k, l)$. Let $(\phi_n)_{n\in\mathbb{N}}$ be a computable enumeration of Turing machines.

In a segment at time $t_i, i \in \mathbb{N}$ with $f(i) = (j, k, l)$, the idea is to simulate the computation of ϕ_j on some sequence of consecutive inputs and fill the writing layer of the segment differently whether the machine halts on each input in this sequence or not. We will say that the sequence is successful if the machine does halt on each input. Indeed saying that the machine is co-finite means that there exists $k \in \mathbb{N}$ such that all sequences $\{k, k+1, \ldots, k+l\}$ are successful. Thus, we will write a testimony (taking the form of a prefix of $(\boxed{\$_0}\boxed{\$}^k)^{\mathbb{N}}$) of the success of a sequence starting at k. We will have to avoid writing a testimony for k more than once for each l.

More formally, the computation will be the following :

- Compute $(j, k, l) = f(i)$ and fill the writing layer of s with $\boxed{\$}^{|s|}$.
- Compute $i_0 = \max\{i' < i, f(i') = (j, k, l)\}$.
- At the same time, simulate the machine ϕ_j with successive inputs $k, k+1, \ldots, k+l$. If one of these simulations does not halt, then stop the simulation after $K^i/2$ steps. In this case, the triple (j, k, l) is said to be failed at time t_i.
- If the machine ϕ_j does halt on all these inputs before timestep $K^i/2$, then denote τ and σ the exact time and space used for the whole computation.
- If $\tau \leq K^{i_0}/2$ and $\sigma \leq i_0$, the whole computation already finished in segments at time K^{i_0} so we do nothing and the segment is said to be failed again.
- In the remaining case, write a prefix of $(\boxed{\$_0}j\boxed{\$_1}k\boxed{\$_2})^{\mathbb{N}}$ in the writing layer of s. This takes less than $K^i/2$ timesteps for acceptable segments. In this case, the triple is said to be successful at time t_i.

Note that as the space used for computation depends only on i, the result of the computation is the same for all segments at some given time.

Consider the word $u_j = \boxed{\$_0}j\boxed{\$_1}$, we will prove that :

$$u_j \in L_\mu(\mathcal{A}) \Leftrightarrow \phi_j \in COF$$

Claim 4.5. $\phi_j \in COF \Rightarrow u_j \in L_\mu(\mathcal{A})$

Proof. First suppose that $\phi_j \in COF$ for some $j \in \mathbb{N}$. In this case, there exists $k \in \mathbb{N}$ such that $\forall l \geq k$, ϕ_j halts on input l. This means that for any $(j, k, l), l \in \mathbb{N}$, there exists $i \in \mathbb{N}$ such that $f(i) = (j, k, l)$ and $K^i/2$ timesteps are enough to simulate ϕ_j on inputs $k, k+1, \ldots, k+l$ and verify that it halts in each case. Thus every triplet $(j, k, l), l \in \mathbb{N}$ is successful for some $i_l \in \mathbb{N}$.

Hence, at time $t_{i_l+1} - 1$, a prefix of $(\boxed{\$_0}j\boxed{\$_1}k\boxed{\$_2})^{\mathbb{N}}$ is written in every acceptable segment. For l, and thus i_l, large enough, the density of the word u_j in every $w_{t_{i_l+1}-1}(s)$ for every acceptable segment is larger than $\frac{1}{2}\frac{1}{|j|+|k|}$ which is a constant. The Lemma 3.7 allows us to conclude that $u_j \in L_\mu(\mathcal{A})$.

□

Claim 4.6. $\phi_j \notin COF \Rightarrow u_j \notin L_\mu(\mathcal{A})$

Proof. Here, with j fixed, if we take an infinite number of successful sequences (j,k,l), they necessarily concern unbounded values of the starting point k. We will use the fact that the density of u_j decreases when k increases.

Suppose $\phi_j \notin COF$ for $j \in \mathbb{N}$. Take $\varepsilon > 0$. For any $i \in \mathbb{N}$, if $f(i) = (j', k, l)$ with $j \neq j'$, then for any acceptable segment s at time t_i, u_j is never written in the segment and $d_{w_{t_{i+1}-1}(s)}(u_j) = 0$.

There exists $k_0 \in \mathbb{N}$, such that $\frac{1}{|j|+|k_0|} < \varepsilon$. As ϕ_j is not co-finite, there exists $l_0 \geq k_0$ such that ϕ_j does not halt on input l_0. Thus, there are at most ${l_0}^2$ triplets $(j,k,l) \in \mathbb{N}^3$ with $k \leq k_0$ and $l \leq l_0$. There exists $i_0 \in \mathbb{N}$ such that for any triplet $(j,k,l), k \leq k_0, l \leq l_0$:

- either (j,k,l) is never successful;
- or there exists $i < i_0$ such that (j,k,l) is successful at time i.

Now consider an acceptable segment s at time $i \geq i_0$.

- If $f(i) = (j', k, l), j' \neq j$ then we have $d_{w_{t_{i+1}-1}(s)}(u_j) = 0$.
- If $f(i) = (j,k,l), k \geq k_0$ then $d_{w_{t_{i+1}-1}(s)}(u_j) \leq \frac{1}{|j|+|k_0|} < \varepsilon$.
- If $f(i) = (j,k,l), k \leq k_0$, then $d_{w_{t_{i+1}-1}(s)}(u_j) = 0$ since $i \geq i_0$.

Finally, we can use the second case of Lemma 3.7 to conclude, and we proved that if ϕ_j is not co-finite, $u_j \notin L_\mu(\mathcal{A})$. □

□

4.3. Descriptive complexity

In this section we will use Lemma 3.7 to construct cellular automata whose μ-limit sets are constrained to be in a specific subshift. We have the following proposition:

Proposition 4.7. *Given a non-empty effective subshift S over an alphabet Σ, there exists a CA whose μ-limit set is included in S.*

Proof. Because the subshift S is effective, it can be characterized by a recursively enumerable set of forbidden words. We will use Lemma 3.7 with a cellular automaton who enumerates forbidden words and fills its segments with words containing none of them.

The behavior of a segment of length n is as follows :

- After computing its own length it starts enumerating and storing forbidden words of S. This enumeration is allowed to use at most a space of $\log n$ cells. When the space is exhausted, the enumeration stops and the computed forbidden words are used for the next steps;

- All possible words of length $\log n$ over Σ are then enumerated in lexicographical order until one is found that does not contain any of the forbidden words previously enumerated (there exists one because the subshift is non-empty);
- The $(\log n)$-long word containing no previously-found forbidden word is copied $n/\log n$ times to fill the whole segment, using a special $\boxed{\$}$ symbol as a delimiter between copies. All temporary computations are erased and the segment remains in this state until it merges with another one at which point the whole process starts again on a longer segment.

We now want to apply Lemma 3.7 to prove that the μ-limit set of this automaton contains only configurations in S by showing that the μ-persistent language contains only words in Σ^* and none of the forbidden words.

We first prove that the computations performed by the segment take at most $O(\log n)$ space and $O(n)$ time. Computing the length of the segment can be done in linear time with a logarithmic set of computing cells that move through the segment. Enumerating forbidden words and then finding a word of length $\log n$ that contains none is done in logarithmic space too. Finally, copying the word that was found can also be done in linear time using a logarithmic set of computing cells that move across the segment. Moreover the $\boxed{\$}$ states inserted between copies appear with density $O(\log n/n)$ on a segment of length n.

According to Lemma 3.7, all this means that the μ-persistent language of the automaton contains only words in Σ^*.

Now let us consider a forbidden word w in the recursively enumerable set that characterizes the subshift S. Given enough space a segment is bound to enumerate w. This means that all long enough segments will enumerate w during their initial enumeration computation and that w cannot be a factor of the $(\log n)$-long word that fills any of those long enough segments. The word w can therefore not be in the persistent language either.

This all means that the μ-persistent language of the automaton is included in Σ^* and contains none of the forbidden words of S, which means that all configurations in the μ-limit set of the automaton are in S. □

This proposition does not allow to describe the μ-limit set obtained, except if the subshift is minimal. A subshift is said to be *minimal* ([LM95]) when it does not contain a proper subshift. Hence the proposition implies that :

Corollary 4.8. *Given a non-empty minimal effective subshift S, there exists a cellular automaton whose μ-limit set is S.*

We will see now how the previous proposition implies the existence of a cellular automaton whose μ-limit set contains only configurations of high Kolmogorov complexity.

Definition 4.2. Given a recursive function $f : \{0,1\}^* \to \{0,1\}^*$, the Kolmogorov complexity relative to f of a string $x \in \{0,1\}^*$ is defined as $K_f(x) = \min\{|y|,\ f(y) = x\}$.

As such, the definition of Kolmogorov complexity depends heavily on the choice of the function f and it is not properly defined for words x such that $\{|y| \mid f(y) = x\}$ is empty. However, it can be shown that there exists a recursive function U such that, for any recursive function f, there is a constant $c_f \in \mathbb{N}$ such that for any string $x \in \{0,1\}^*$ we have $K_U(x) \leq K_f(x) + c_f$. This also implies that $K_U(x)$ is properly defined for all x. The Kolmogorov complexity of a string x is then defined as $K(x) = K_U(x)$ for some such *additively optimal* U.

Informally, the Kolmogorov complexity of a word is the length of its shortest possible description.

Definition 4.3 (α-complexity)**.** Given a constant $\alpha > 0$, a word of length n on the alphabet $\{0,1\}^*$ is said to be *α-complex* if its Kolmogorov complexity is greater than αn. A word that is not α-complex is said to be *α-simple*.

Corollary 4.9 (of Proposition 4.7)**.** *For any $\alpha < 1$, there exists a constant n_α and a cellular automaton whose μ-limit set contains only configurations whose factors of length greater than n_α are all α-complex.*

Proof. To use Proposition 4.7 we need to show that for some n_α the subshift of configurations over $\{0,1\}$ that contain no α-simple word of length greater than n_α is effective and non-empty.

As for the effectiveness, a word x is α-simple if and only if there exists y such that $U(y) = x$ and $|y| \leq \alpha|x|$. We can enumerate all such words by dovetailing the computations of $U(y)$ for all possible y and checking if the resulting word is α-simple by comparing its length to that of the input y. Therefore the set of α-simple words $\{x,\ K(x) \leq \alpha|x|\}$ is recursively enumerable, and so is the set of such words of length greater than n_α.

The existence of n_α and a configuration containing no α-simple factor of length greater than n_α is a consequence of the main result in [RU06] since there exist at most $2^{\alpha n}$ forbidden words of length n and complexity less than αn. □

Corollary 4.10 (of Corollary 4.9)**.** *There exists a CA whose μ-limit set contains only non-recursive configurations.*

Proof. In a recursive configuration c, the word $c_{[0,n]}$ starting at position 0 and of length n has complexity $O\big(\log(n)\big)$. Therefore no recursive configuration can be α-complex in the sense defined above. Corollary 4.9 concludes the proof. □

As a last application of Proposition 4.7, we will show that the quasi-periodicity of a μ-limit set can be highly non-trivial using a result of [BJ10]. A configuration c is said *quasi-periodic* if any pattern occurring in c occurs in any large enough pattern of c. Any subshift contains a quasi-periodic configuration [Bir12]. For such configurations the quasi-periodicity can be quantified through the *quasi-periodicity function*.

Definition 4.4. Let c be a quasi-periodic configuration. We associate to c the *quasi-periodic function* $\rho_c : \mathbb{N} \to \mathbb{N}$ defined by:

$$\rho_c(n) = \max_{u \in L(c), |u|=n} \min\{p : \text{any pattern of size } p \text{ of } c \text{ contains } u\}$$

Corollary 4.11 (of Proposition 4.7)**.** *There exists a cellular automaton such that for any quasi-periodic configuration c of its μ-limit set, the function ρ_c can not be bounded by any recursive function.*

Proof. It is a direct application of Proposition 4.7 with the effective subshift obtained by corollary 3.4 of [BJ10]. □

5. Complexity of properties of μ-limit sets

5.1. A Rice theorem for μ-limit sets

In the case of the limit set of cellular automata, J. Kari[Kar94] proved a result equivalent to Rice theorem, meaning that any non trivial property of limit sets of cellular automata is undecidable. Using certain aspects of his technique, we will prove here that any non trivial property of μ-limit sets of cellular automata has a higher complexity than the negation of the problem of being co-finite for a Turing machine. Since we will deal with different cellular automata in this section, the considered measures will be the uniform ones on each alphabet.

5.1.1. Properties of μ-limit sets

Intuitively, a property of the μ-limit set is a property $\mathcal{P}$ which depends only on the μ-limit set: if two CA have the same μ-limit set, then either both have property $\mathcal{P}$ or none has property $\mathcal{P}$. We use the same formalism as J. Kari for limit sets. Consider a countable set $X_\alpha = \{\alpha_0, \alpha_1, \dots, \alpha_n, \dots\}$ of possible states, that is, we will consider cellular automata whose sets of states is a finite subset of X_α.

Definition 5.1. A property of μ-limit sets of cellular automata is a family $\mathcal{P} \subseteq \mathfrak{P}(X_\alpha^{\mathbb{Z}})$. A μ-limit set of some cellular automaton is said to have property $\mathcal{P}$ if it is included in $\mathcal{P}$.

For example, μ-nilpotency is given by the family $\{\alpha_i^{\mathbb{Z}}, i \in \mathbb{N}\}$. We will talk equivalently of properties of μ-limit sets and μ-limit languages, but a property of cellular automata *concerning* the μ-limit set is not necessarily a property *of* μ-limit sets. The surjectivity is the classical example to show that both differ. Indeed surjectivity refers to the set of states of the automaton and not necessarily only to those appearing in the μ-limit set. Note also that there is no obvious relationship between properties of μ-limit sets and properties of limit sets:

- nilpotency is a property of limit sets but not a μ-limit property (*e.g.* for μ the uniform Bernoulli measure, any CA with a spreading state is μ-nilpotent but can be nilpotent or not);
- conversely, μ-nilpotency is a property of μ-limit sets, but it is not known whether it is a property of limit sets.

A property is said to be *trivial* when either it contains all μ-limit sets or none.

5.1.2. Computing a weakly generic configuration

In order to prove this Rice theorem, we will need to be able to compute the prefixes of some weakly generic configuration, we will then refer to the following proposition proved in [FK77]:

Proposition 5.1. *There exists a weakly generic configuration c_G on the finite alphabet X such that :*

- *there exist $A, B > 0$ such that for any $l \in \mathbb{N}$, $u \in X^l$ and $L \geq |X|^{2l}$, we have :*
$$A|X|^{-l} \leq d_{c_G[0,L-1]}(u) \leq B|X|^{-l}$$
- *the prefix of size n of c_G can be computed in $O(\log n)$ space and $O(n)$ time.*

Remark 5.1. The property over the densities of prefixes can be extended to images of c_G by a cellular automaton, for $k \in \mathbb{N}$ and $L \geq 2k$:

$$A/2d_{\mathcal{A}^k(c_G)}(u) \leq d_{\mathcal{A}^k(c_G)_{[0,L-1]}}(u) \leq 2Bd_{\mathcal{A}^k(c_G)}(u)$$

5.1.3. Construction

Theorem 5.2. *Given a property $\mathcal{P}$ of μ-limit sets, either $\mathcal{P}$ is trivial or $\mathcal{P}$ is Π_3-hard.*

To prove this theorem, we will use a reduction to the problem of being co-finite for a Turing machine which is Σ_3^0-complete.

The general idea of the proof is close to what J. Kari did for limit sets, using the following proposition :

Proposition 5.3. *Given a cellular automaton $\mathcal{A}$ and a Turing machine ϕ, there exists a cellular automaton $\mathcal{B}$ such that :*

- *if $\phi \in COF$ then $\Lambda_\mu(\mathcal{B}) = \Lambda_\mu(\mathcal{A})$;*
- *else $\Lambda_\mu(\mathcal{B}) = Q_{\mathcal{A}}^{\mathbb{Z}}$.*

Using this property, whose proof will follow, we can prove Theorem 5.2.

Proof. Given some non trivial property $\mathcal{P}$ of μ-limit sets, consider a Turing machine ϕ and cellular automata $\mathcal{A}_1$ and $\mathcal{A}_2$ such that exactly one among $\mathcal{A}_1$ and $\mathcal{A}_2$ has property $\mathcal{P}$. We consider they have a common alphabet, which is always possible by increasing their alphabets if necessary. Suppose there is an algorithm with oracle to decide $\mathcal{P}$. First denote $\mathcal{B}_1$ and $\mathcal{B}_2$ the cellular automata given by Proposition 5.3 for respectively ϕ and $\mathcal{A}_1$ and ϕ and $\mathcal{A}_2$. Then using our algorithm on $\mathcal{B}_1$ and $\mathcal{B}_2$, we can decide if the answer is the same or not. The first case corresponds necessarily to $\phi \notin COF$ and the second to $\phi \in COF$. Thus, it is possible to decide with the same oracle whether ϕ is co-finite. This means that the oracle is at least Σ_3-hard.

□

Now we prove Proposition 5.3

Proof. The proof will have similarities with the one of Theorem 4.4

The CA $\mathcal{B}$ has the global structure of the automaton described in Section 3. Let us describe the computation inside segments. First consider a computable enumeration f of $\mathbb{N}^2$, such that for any $(k,l) \in \mathbb{N}^2$ there exist infinitely many $i \in \mathbb{N}$ with $f(i) = (k,l)$.

In a segment, the idea is to simulate the computation of ϕ on some sequence of consecutive inputs and fill the writing layer of the segment differently whether the machine halts on each input in this sequence or not. We will say that the sequence is successful if the machine does halt on each input. Indeed saying that the machine is co-finite means that there exists $k \in \mathbb{N}$ such that all sequences $\{k, k+1, \dots, k+l\}$ are successful. Thus, we will write a testimony of the success of a sequence starting at k. We will have to avoid writing a testimony for k more than once for each l.

More formally, the computation will be the following in a segment s at time t_i :

- Compute $(k,l) = f(i)$ and $\nu(i) = \lfloor \log i \rfloor$.
- Compute $i_0 = \max\{i' < i, f(i') = (k,l)\}$.
- At the same time, simulate the machine ϕ with successive inputs $k, k+1, \dots, k+l$. If one of these simulations does not halt, then stop the simulation after $K^i/2$ steps. In this case, the couple (k,l) is said to be failed at time t_i.
- If the machine ϕ does halt on all these inputs before timestep $K^i/2$, then denote τ and σ the exact time and space used for the whole computation.
- If $\tau \leq K^{i_0}/2$ and $\sigma \leq i_0$, the whole computation necessarily already finished in segments at time K^{i_0} and again the couple (k,l) is said to be failed at time t_i.
- In failed segments, compute and write $\mathcal{A}^{\nu(i)}(c_G)_{[0,|s|-1]}$ in the writing layer of the segment.
- In the remaining case, compute $u_i = c_{G[0,\nu(i)]}$ and $v_i = \mathcal{A}^{\nu(i)}(c_G)_{[0,\nu(i)]}$. Then write a prefix of $(u_i v_i^k)^{\mathbb{N}}$ in the writing layer of s. In this case, the couple is said to be successful at time t_i.

It is clear thanks to Proposition 5.1 that for i large enough the computation is finished before $t_{i+1} - 1$. Note as well that as the space used for computation depends only on i, the result of the computation is the same for all segments at some given time.

Clearly, thanks to Lemma 3.7, $L_\mu(\mathcal{B}) \subseteq Q_{\mathcal{A}}^{\mathbb{Z}}$.

Claim 5.4. *If $\phi \in COF$ then $\Lambda_\mu(\mathcal{B}) = Q_{\mathcal{A}}^{\mathbb{Z}}$.*

Proof. In this case, there exists $k \in \mathbb{N}$ such that $\forall l \geq k$, ϕ halts on input l. This means that for any $(k,l), l \in \mathbb{N}$, there exists $i \in \mathbb{N}$ such that $f(i) = (k,l)$ and $K^i/2$ timesteps are enough to simulate ϕ on inputs $k, k+1, \ldots, k+l$ and verify that it halts in each case. Thus every couple $(k,l), l \in \mathbb{N}$ is successful at some time t_{i_l}.

Hence, at time $t_{i_l+1}-1$, in every acceptable segment, a proportion $1/k$ of the segment is filled with some prefix of c_G. Hence, for l, and thus i_l, large enough, the density of any word $u \in Q_{\mathcal{A}}^*$ in every $w_{t_{i_l+1}-1}(s)$ of an acceptable segment s is larger than $\frac{A}{2k|Q_{\mathcal{A}}|^{|u|}}$ (with A from Proposition 5.1) which is a constant as k is fixed. The Lemma 3.7 allows us to conclude that $u \in L_\mu(\mathcal{B})$.

□

Claim 5.5. *If $\phi \notin COF$ then $\Lambda_\mu(\mathcal{B}) = \Lambda_\mu(\mathcal{A})$.*

Proof. Here, if we take an infinite number of successful sequences, they necessarily concern unbounded values of the starting point k. We will use the fact that the space covered by prefixes of an image of c_G decreases when k increases.

Take $\varepsilon > 0$.

There exists $k_0 \in \mathbb{N}$, such that $\frac{1}{k_0} < \varepsilon$. As ϕ is not co-finite, there exists $l_0 \geq k_0$ such that ϕ does not halt on input l_0. There are less than ${l_0}^2$ couples $(k,l) \in \mathbb{N}^2$ with $k \leq k_0$ and $k+l \leq l_0$. Denote i_0 the smallest integer such that for each such (k,l) :

- either ϕ does not halt on some input between k and $k+l$;
- or there exists $i' \leq i_0$ such that ϕ halts on all these inputs within space i' and total time $K^{i'}/2$.

Thus, when $f(i) = (k,l)$ with $i \geq i_0$ and $k \leq k_0$, the couple (k,l) is failed and the writing layer of s contains a prefix of $\mathcal{A}^{\nu(i)}(c_G)$.

Now consider an acceptable segment s at time $i \geq i_0$ with $f(i) = (k,l)$.

- If $k \geq k_0$ then (k,l) is successful but the space covered by copies of u_i is less than $\frac{2}{k_0} < 2\varepsilon$.
- If $k \leq k_0$, then (k,l) is failed and the writing layer of s contains a prefix of $\mathcal{A}^{\nu(i)}(c_G)$.

Finally, thanks to Remark 5.1 and the second case of Lemma 3.7, we conclude that $L_\mu(\mathcal{B}) = L_\mu(\mathcal{A})$.

□

□

As a consequence of this proof, we have that :

Proposition 5.6. *If Σ is the μ-limit set of a CA, then it is the μ-limit set of a CA using the design of the construction of Section 3.*

Indeed, to prove the theorem, we used a simulation of some CA inside segments, showing that it did not modify the μ-limit set. It can be achieved for any CA, thus any μ-limit set can be reached this way.

In the next section, we will deal more specifically with μ-nilpotency. We leave open the question of properties of higher complexity. For example, being a shift of finite type, a sofic shift or containing a weakly generic configuration... In particular, it is not known whether there exist properties of arbitrary high complexity.

5.2. μ-nilpotency

Recall that a CA is μ-nilpotent if and only if its μ-limit set is a singleton.

Proposition 5.7. *Let μ be a computable measure. The set of μ-nilpotent CA is Π_3^0.*

Proof. If a CA is μ-nilpotent then the only configuration in the μ-limit set is necessarily of the form ${}^\omega q^\omega$ for some state q. Hence, μ-nilpotency is equivalent to the following property:

$$\forall \varepsilon > 0, \exists t_0, \forall t \geq t_0, \exists q_0,\ F^t\mu(q_0) \geq 1 - \varepsilon$$

Since μ is computable and the number of state of a CA is finite, the predicate "$\exists q_0, F^t\mu(q_0) \geq 1 - \varepsilon$" (depending on t, F and ε) is recursive, which concludes the proof. □

The following theorem is a direct consequence of the Rice Theorem (5.2) proved earlier.

Theorem 5.8. *Let μ be the uniform measure on the fullshift, the problem of being μ-nilpotent for a cellular automaton is Π_3^0-complete.*

Proposition 5.9. *Let μ be a σ-ergodic measure of full support. Then we have:*

- *the set of μ-nilpotent CA with a persistent state is co-recursively enumerable;*
- *the set of μ-nilpotent CA with equicontinuous points is Σ_2^0.*

Proof. Using Proposition 4.1, not being μ-nilpotent is equivalent to the existence of different words of same size in the temporal period of some spatially periodic configuration containing a wall. For CA with a persistent state, it is sufficient to test with a wall made of r adjacent persistent states (r being the radius). Hence we can recursively enumerate CA with a persistent state and a pair of different words as said above. The first item of the Proposition follows.

For CA with equicontinuous points, the additional difficulty is that we don't know *a priori* which word is a wall. Testing this costs an additional quantifier. Formally, a CA is μ-nilpotent with an equicontinuous point if and only if

$$\exists w, q_0 \big(\forall z, t\ R(w, z, t)\ \wedge\ \forall v\ R'(q_0, w, v)\big)$$

where predicates R and R' are recursive and such that:

- $R(w, z, t)$ checks that w is a wall up to time t and position z and $-z$ (see definition in Section 4.1)
- $R'(q_0, w, v)$ checks that periodic configuration wv converges to the q_0-uniform configuration (exponential time bound is enough to check)

The second item of the Proposition follows. □

The definition of μ-nilpotency has been chosen analogously as the definition of nilpotency. But in the case of nilpotent CA, we can show that the limit set contains either a unique uniform configuration or an infinite number of distinct configurations. As this property is false for μ-limit sets, a notion of *weak μ-nilpotency* can be defined. The most natural way is to say a CA is weakly μ-nilpotent when its μ-limit set is finite. Still, some refinements can be considered, such as μ-limit sets containing only uniform configurations or the shift-orbit of one unique periodic configuration.

In terms of complexity, the alphabet being finite, the second definition (only uniform configurations) is equivalent to classical μ-nilpotency. Thanks to Rice Theorem, other ones are at least as complex, but we need other quantifiers to describe the finite μ-limit set.

6. Types of convergence towards the limit

6.1. Simple convergence

By definition words which are not in the μ-limit language are those whose probability goes to zero as time increases. However, this probability does not always converge for words which are in the μ-limit language. As a consequence, contrary to the limit set, the μ-limit set is generally changed when taking iterates of a given CA.

Theorem 6.1. *There exist F such that F and F^2 do not have the same μ-limit set.*

Proof. We consider the uniform Bernoulli measure. To construct such a F it is sufficient to use the counter construction from Section 3, *i.e.* the initialization step of our main construction techniques. We just use the trick of unary counters to build a growing uniform "protected area" alternating between two states: all black (odd steps), or all white (even steps). We keep the same collision rule described in Section 3:

- when two areas of different ages collide, the older is destroyed by the younger;
- when two areas of same age collide, they simply merge (it is possible since, having the same age, they have the same uniform content).

We say a cell is *synchronized* at time t_0 if for any $t \geq t_0$ it is **black** when t is odd and **white** when t is even. Then, using a simplified version of Lemma 3.2, we can prove the following:

Claim. Starting from a generic configuration, the density of cells which are synchronized at time t goes to 0 when t grows.

It follows that F^2 is μ-nilpotent whereas the μ-limit set of F contains two configurations: the "all black" and the "all white". □

We say that a CA is *simply convergent for* μ if the probability of appearence of a word u converges for any u, *i.e.*

$$\forall u \in A^*, \exists \alpha \in \mathbb{R}, \forall \varepsilon > 0, \exists t_0, \forall t \geq t_0 : \ \left|\mathcal{A}^t \mu([u]_0) - \alpha\right| \leq \varepsilon.$$

Examples of simply convergent CA are μ-nilpotent CA. Indeed, the probability of apparence of any word goes to 0 except for one word of each size for which it necessarily goes to 1.

If F is simply convergent for μ then, for any $t \geq 1$, F^t is simply convergent and F and F^t have the same μ-limit set. The Theorem 6.1 above gives an example of CA which is not simply convergent.

As shown by the following theorem, the simple convergence asumption simplifies the μ-limit set as well as some decision problems on it (to be compared to Theorems 4.4 and 5.8).

Theorem 6.2. *Let μ be a computable translation invariant measure.*

- *if $\mathcal{A}$ is simply convergent for μ then $L_\mu(\mathcal{A})$ is a Σ_2^0 set;*
- *there exists a Π_2^0 predicate that characterizes μ-nilpotent CA among simply convergent CA;*
- *the set of simply convergent CA is Π_3^0 and it is Pi_3-hard when μ is the uniform bernouilli measure.*

Proof. If $\mathcal{A}$ is simply convergent for μ, we have the following characterization of L_μ:

$$u \in L_\mu \Leftrightarrow \exists t_0, \exists \varepsilon, \forall t > t_0 : \ F^t \mu([u]_0) > \varepsilon.$$

We deduce the first item of the theorem.

$\mathcal{A}$ is not μ-nilpotent exactly when there are two different words of equal size in L_μ. With the hypothesis of simple convergence, it can be written:

$$\exists u, v, |u| = |v|, u \neq v, \exists t_0, \exists \varepsilon_u, \exists \varepsilon_v, \forall t > t_0 : \ F^t \mu([u]_0) > \varepsilon_u \wedge F^t \mu([v]_0) > \varepsilon_v$$

and the second item of the theorem follows directly.

To show the third item, let us first remark that simple convergence can be expressed by a Π_3^0 formula saying that the sequence of probabilities of apparence along time of each word is a Cauchy sequence:

$$\forall u \in A^*, \forall \varepsilon > 0, \exists N, \forall p, q > N : \ \left|\mathcal{A}^p \mu([u]_0) - \mathcal{A}^q \mu([u]_0)\right| \leq \varepsilon.$$

Finally, for Π_3^0-hardness it is sufficient to verify that a subset of the CA constructed in the proof of Proposition 5.3 are either μ-nilpotent (hence simply

convergent), or not simply convergent. More precisely, in the construction, consider a μ-nilpotent CA $\mathcal{A}$ ($L_\mu(\mathcal{A}) = \boxed{\$}^*$ for some special state $\boxed{\$}$). If you take a machine $\phi \notin COF$, then $\mathcal{B}$ is μ-nilpotent as shown in Claim 5.5. In the other case, with $\phi \in COF$, we still have $\limsup_t F^t\mu([\boxed{\$}]_0) = 0$. Indeed, you get the result with a sequence of steps $(t_j)_j$ where $f(j) = (j, 0)$. Hence, as in this case $L_\mu(\mathcal{A}) = Q^*_{\mathcal{A}}$ (Claim 5.4), the convergence cannot be simple.

□

Complexity considerations allows to prove that some μ-limit sets are impossible to obtain with simple convergence. We currently do not know any direct proof of this fact.

Corollary 6.3. *There exists a CA whose μ-limit set can not be the μ-limit set of any simply convergent CA.*

Proof. By Theorem 4.4 there exists a Σ^0_3-hard μ-limit set. However, Theorem 6.2 shows that simply convergent CA produce μ-limit sets which are only Σ^0_2. □

6.2. Cesaro mean

Other definitions could be considered for μ-limit sets, in particular the Cesaro mean could be used.

Definition 6.1 (Cesaro-persistent set)**.** For a CA $\mathcal{A}$, we define the *Cesaro-persistent set* $C_\mu(\mathcal{A}) \subseteq Q^*$ by: $\forall u \in Q^*$:

$$u \notin C_\mu(\mathcal{A}) \iff \lim_{n\to\infty} \sum_{k\leq n} \mathcal{A}^k\mu([u]_0) = 0.$$

Then the *μ-Cesaro-limit set* of $\mathcal{A}$ is $\Lambda C_\mu(\mathcal{A}) = \left\{c \in Q^{\mathbb{Z}} : L(c) \subseteq C_\mu(\mathcal{A})\right\}$.

All the theorems proved thanks to Lemma 3.7 still hold with Cesaro mean. Indeed, the second point is unchallenged by the change of definition, and for the first point, we obtain the result with a slight modification of the proof. Take $K = 3$ for example and at time $t_i, i \in \mathbb{N}$, allow K^i timesteps for computation and do nothing for K^i more timesteps. Thus, Lemma 3.7 holds with Cesaro mean, as well as all further constructions. In this case μ-Cesaro limit set and μ-limit set are equal, and it proves (with Proposition 5.6) that any μ-limit set can be the μ-Cesaro-limit set of a CA.

We can also prove that any μ-Cesaro-limit set of a CA is the μ-limit set of another one. The idea of the proof is similar to the one of Proposition 5.6, but instead of filling segments with some part of the image of a weakly generic configuration, we fill it with a concatenation of parts of the first images of this configuration : $(\mathcal{A}^k(c_G))_{k\leq\nu(i)}$ at time t_i with $\nu(i)$ small enough. We hence have :

Proposition 6.4. *For μ the uniform Bernoulli measure, the sets of μ-limit sets and μ-Cesaro-limit sets are equal.*

Whether the μ-limit and μ-Cesaro-limit sets of a given CA are equal is a natural question and as we will see with the help of a previous example, the answer is negative. Clearly ΛC_μ is included in Λ_μ for any CA but the converse is false. Indeed if we consider the automaton used to prove 4.4, and decide to write over the segment only at the end of the computation time, the Cesaro mean will erase the result of the computation from its μ-Cesaro-limit set. Thus :

Proposition 6.5. *There exists a cellular automaton whose μ-language is Σ_3-complete and which is μ-Cesaro nilpotent.*

6.3. Non-recursive convergence time

Here we want to point out the fact that convergence to the μ-limit language may actually be really late, in particular the next proposition states that the convergence rate may be slower than any recursive function.

Proposition 6.6. *Given an enumeration of Turing machines, denote $T_m(k)$ the halting time of machine $m \in \mathbb{N}$ on input $k \in \mathbb{N}$. If m does not halt on k, $T_m(k) = 0$.*

There exists a cellular automaton F (with $0 \in Q_F$) such that :

- $\forall n \in \mathbb{N}, \exists t_n \geq \max_{k,m} \{T_m(k) : k \leq n, m \leq n\}, F^{t_n}\mu([0]) \geq \frac{1}{2n}$;
- $0 \notin L_\mu(F)$.

Proof. To prove it we use again the construction of Section 3, and nearly the same construction as described in the proof of Theorem 4.4. We consider at time $t_i, i \in \mathbb{N}$ some couple (instead of a triple) $f(i) = (j,k) \in \mathbb{N}^2$ given by the enumeration f, and try to finish the computation of machine j with input k. Each couple (j,k) is said to be successful for the smallest i such that the computation halts, and failed otherwise. In the particular case of a successful couple, we now write a prefix of the configuration $\left(1^{\max(j,k)-1}0\right)^{\mathbb{N}}$. Each time a couple is failed, the segment is filled with 1.

Thus 0 has density $\frac{1}{n}$ in the writing layers of segments only when the simulation of the machine halts for the first $i = f(j,k)$ with $n = \max\{j,k\}$, and 0 otherwise. The two points of the result are now easily verified.

For the first point, given $n \in \mathbb{N}$, consider (k_0, m_0) such that $T_{m_0}(k_0) = \max_{k,m} \{T_m(k) : k \leq n, m \leq n\}$. When (k_0, m_0) is successful, the density of the word 0 in acceptable segments is larger than $\frac{1}{n}$. For n large enough, these segments cover enough space to let us conclude.

For the second point, given $n \in \mathbb{N}$, there exists $i_0 \in \mathbb{N}$ such that after i_0, acceptable segments have enough time to finish the computation of any machine $j \leq n$ with input $k \leq n$, if it ever halts. Then, there is $i_1 \geq i_0$ such that all couples $(j \leq n, k \leq n)$ have been enumerated between i_0 and i_1. Thus, after i_1 the density of the word 0 is less than $\frac{1}{n}$ in these segments.

□

7. Recap of results

In this section μ denotes the uniform Bernoulli measure. First we give comparative recap of complexity of properties or problems concerning limit sets and μ-limit sets.

Problem or property	**Limit Set**	**μ-Limit Set**
Being a singleton	Σ^0_1-complete (see [Kar92])	Π^0_3-complete (see Thm. 5.8)
Any non-trivial property	Σ^0_1-hard (see [Kar94])	Π^0_3-hard (see Thm. 5.2)
Worst-case language	Π^0_1-complete (see [Hur87])	Σ^0_3-complete (see Thm. 4.4)
Simplest configuration	always uniform	can be α-complex (see Cor. 4.9)
Simplest quasi-periodicity	always periodicity	can be not recursively bounded (see Cor. 4.11)

Below is a recap on how the complexity of some problems is affected by adding hypothesis on the input CA.

Type of input CA	**Worst L_μ**	**μ-Nilpotency**
General case	Σ^0_3-complete (see Thm. 4.4)	Π^0_3-complete (see Thm. 5.8)
Equicontinuous	Σ^0_1 (see Thm. 4.2)	Σ^0_2 (see Prop. 5.9)
Simply convergent	Σ^0_2 (see Thm. 6.2)	Π^0_2 (see Thm. 6.2)

We left some questions open in previous sections, but a few general ones remain. As shown in [dMS13] it is certainly possible to generalize the results obtained here for large sets of measures (which was not the purpose of the present paper). In this context, it becomes relevant to consider the particular case of surjective CA. Indeed, as the uniform bernoulli measure is preserved by surjective CA, the μ-limit set is the fullshift, but for another measure, the question is open.

Naturally, the extension of these results can be discussed for higher dimensions. In particular, some of them should be reached given an equivalent construction in higher dimensions.

References

[BDS10] Laurent Boyer, Martin Delacourt, and Mathieu Sablik. Construction of μ-limit sets. In *JAC*, pages 76–87, 2010.

[Bir12] G. D. Birkhoff. Quelques théor'emes sur le mouvement des systèmes dynamiques. *Bulletin de la Société Mathématique de France*, 1912.

[BJ10] Alexis Ballier and Emmanuel Jeandel. Computing (or not) quasi-periodicity functions of tilings. In *JAC*, pages 54–64, 2010.

[BPT06] 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.

[Del11] Martin Delacourt. Rice's theorem for -limit sets of cellular automata. In *ICALP (2)*, pages 89–100, 2011.

[dMS13] 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.

[FK77] Harold Fredricksen and Irving J. Kessler. Lexicographic compositions and debruijn sequences. *J. Comb. Theory, Ser. A*, 22(1):17–30, 1977.

[FK07] Enrico Formenti and Petr Kůrka. A search algorithm for the maximal attractor of a cellular automaton. In *STACS*, pages 356–366, 2007.

[Hur87] Lyman P. Hurd. Formal language characterizations of cellular automaton limit sets. *Complex Systems*, 1:69–80, 1987.

[Hur90a] Mike Hurley. Attractors in cellular automata. *Ergodic Theory and Dynamical Systems*, 10:131–140, 2 1990.

[Hur90b] Mike Hurley. Ergodic aspects of cellular automata. *Ergodic Theory and Dynamical Systems*, 10:671–685, 11 1990.

[Kar92] J. Kari. The Nilpotency Problem of One-dimensional Cellular Automata. *SIAM Journal on Computing*, 21:571–586, 1992.

[Kar94] J. Kari. Rice's theorem for the limit sets of cellular automata. *Theoretical Computer Science*, 127:229–254, 1994.

[KM00] P. Kůrka and A. Maass. Limit Sets of Cellular Automata Associated to Probability Measures. *Journal of Statistical Physics*, 100(5-6):1031–1047, 2000.

[Kůr97] Petr Kůrka. Languages, equicontinuity and attractors in cellular automata. *Ergodic Theory and Dynamical Systems*, 17:417–433, 3 1997.

[Kůr03] P. Kůrka. *Topological and symbolic dynamics*. Société Mathématique de France, 2003.

[LM95] D. Lind and B. Marcus. *An introduction to symbolic dynamics and coding*. Cambridge University Press, 1995.

[Maa95] Alejandro Maass. On the sofic limit sets of cellular automata. *Ergodic Theory and Dynamical Systems*, 15:663–684, 7 1995.

[Odi99] Piergiorgio Odifreddi. *Classical Recursion Theory.* Studies in Logic and the Foundations of Mathematics. North Holland, 1999.

[RU06] Andrey Yu. Rumyantsev and M. A. Ushakov. Forbidden substrings, kolmogorov complexity and almost periodic sequences. In *STACS*, pages 396–407, 2006.