μ-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

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

Abstract

This paper concerns μ-limit sets of cellular automata: sets of configurations made of words whose probability to appear does not vanish with time, starting from an initial μ-random configuration. More precisely, we investigate the computational complexity of these sets and of related decision problems. Main results: first, μ-limit sets can have a Σ_3^0-hard language, second, they can contain only α-complex configurations, third, any non-trivial property concerning them is at least Π_3^0-hard. We prove complexity upper bounds, study restrictions of these questions to particular classes of CA, and different types of (non-)convergence of the measure of a word during the evolution.

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 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].

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

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 here as a ready-to-use tool (see Theorem 3.1).
- 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 set contains 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 characterized, 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 shift map, i.e. 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:

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

(in particular $|v|_u = 0$ as soon as $|u| > |v|$).

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 write $|u|_W$ and $d_c(W)$. We can give similar definitions for semi-configurations (indexed by $\mathbb{N}$) too.

We will also use the classical notion of density of a subset $X \subseteq \mathbb{Z}$ of integers and denote it simply d:

$$d(X) = \limsup_{n \to +\infty} \frac{|X \cap \{-n, \dots, n\}|}{2n+1}$$

Definition 2.1 (Growing computable sequence)**.** A sequence $w = (w_i)_{i \in \mathbb{N}}$ of finite words on the alphabet Q is a *growing computable sequence* when:

- $\lim_{i \to \infty} |w_i| = \infty$;
- there exists a Turing machine that computes w_i when given the input i.

Denote $\mathbb{W}(Q)$ the set of growing computable sequences on alphabet Q. For any $w \in \mathbb{W}(Q)$, we define the associated language of persistent words :

$$L_w = \{u \in Q^*, d_{w_i}(u) \not\to_{i \to \infty} 0\}.$$

The following lemma shows that we can produce the persistent language of any given growing computable sequence by another growing computable sequence where we have a precise control on time and space resource needed for the computation of each word of the sequence.

Lemma 2.1. *Let T and S be computable functions from $\mathbb{N}$ to itself which have the following properties:*

- *$T(i) >> i$ and $i >> S(i) >> \log(i)$;*
- *the time complexity of both T and S are $o(T)$;*
- *the space complexity of T is at most S and that of S is $o(S)$.*

Consider any growing computable sequence $w = (w_i)_{i \in \mathbb{N}}$. Then there exists another growing computable sequence $w' = (w'_i)_{i \in \mathbb{N}}$ and a Turing machine ϕ (with possibly several heads and tapes) such that:

- *$L_w = L_{w'}$;*
- *ϕ computes w'_i on input i in time at most $T(i)$ and space at most $S(i)$ (for large enough i).*

Proof. Let ϕ_0 be a Turing machine producing w_j on input j. We can suppose without loss of generality that the time $T_0(j)$ spent by ϕ_0 to produce w_j on input j verifies:

$$T_0(j+1) \geq 2 \cdot T_0(j)$$

This can be obtained by artificially slowing ϕ_0 if necessary (on input j, recompute 2 times step $j-1$ before doing the real work to produce w_j). We now sketch the behavior of ϕ on input i:

- ϕ has an output tape initialized with the empty word;
- it also initializes a space marker at position $S(i)$ and precomputes $\lambda(i-1)$ and $\lambda(i)$ (λ is a function to be precised later, but smaller than T and as easy to compute);
- it simulates ϕ_0 on each successive entry $j \leq i$
- at each step of ϕ_0 it increments some step counter and check that it is less than $\lambda(i)$ and that everything still fits within space $S(i)$:
  - if not it stops with the current output written on the output tape;
  - if it is OK, it goes on;
- when ϕ_0 reaches an halting state on input j, it copies the output produced by ϕ_0 on the output tape and then check that the step counter is less than $\lambda(i-1)$:
  - if it is the case it cleans the working tape of ϕ_0, and start a new simulation of ϕ_0 on input $j+1$;
  - if it is not the case it stops and such outputs $w_j = \phi_0(j)$.

Both the counter incrementation routine and the halting state routine above take time $o(i)$ because it is just a matter of doing a constant number of erasing/copying/comparing/incrementing words of length at most $S(i)$. So if we take $\lambda(i)$ smaller than $(T(i)-i)/i$ we are guaranteed that ϕ halts in time at most $T(i)$ using space at most $S(i)$ (for i large enough). We also want $\lambda(i)$ to grow slowly, precisely such that:

$$\lambda(i+1) < 2 \cdot \lambda(i-1).$$

Then, by construction, ϕ always outputs some w_i or the empty word. First, if $\phi(i)$ produces w_j then by construction $\phi(i+1)$ produces either w_j or w_{j+1}. Indeed if ϕ produces w_j on input i it is because:

- either $\lambda(i-1) < T_0(j)$ and therefore $\lambda(i+1) < 2 \cdot T_0(j) \leq T_0(j+1)$ so that $\phi(i+1)$ also produces w_j;
- or $\lambda(i) < T_0(j+1)$ and therefore $\phi(i+1)$ produces either w_j (in the case where $\lambda(i+1) < T_0(j+1)$) or w_{j+1} (if $\lambda(i+1) \geq T_0(j+1)$).

because $\phi(i)$ produced w_j with step counter at most $\lambda(i)$, but could not produce w_{j+1}, so $\phi(i+1)$ either has time to produce w_{j+1} and outputs that (by the halting conditions) or is short of time and keeps the previous successfull output which is w_j.

Second, for any j, there must be some large enough i such that $\phi(i)$ produces some $w_{j'}$ with $j' \geq j$ (precisely, if i is large enough so that $\phi_0(j)$ halts in less than $\lambda(i)$ steps). Therefore, the sequence w' produced by ϕ is, after some finite prefix of empty words, of the form:

$$\underbrace{w_j, \ldots, w_j}_{\text{finite} > 0}, \underbrace{w_{j+1}, \ldots, w_{j+1}}_{\text{finite} > 0}, \underbrace{w_{j+2}, \ldots, w_{j+2}}_{\text{finite} > 0}, \ldots$$

We deduce that $L_w = L_{w'}$. □

2.2 Cellular Automata

Definition 2.2 (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 denote $\mathcal{A}(c)$ the image of a configuration c given by: $\forall z \in \mathbb{Z}, \mathcal{A}(c)_z = \delta_\mathcal{A}(c_{z-r}, \ldots, c_z, \ldots, c_{z+r})$. 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 write Q, r and δ for $Q_\mathcal{A}$, $r_\mathcal{A}$, $\delta_\mathcal{A}$.

In this paper, to avoid artificial set-theoretical technicalities, we fix some countable set $\mathcal{Q} = \{q_0, q_1, q_2, \ldots\}$ and adopt the convention that all cellular automaton alphabets we consider are subsets of $\mathcal{Q}$. This allows us to speak about the set of all cellular automata, or the set of all sets of configurations.

A state $a \in Q_\mathcal{A}$ is said to be *permanent* for a CA $\mathcal{A}$ if for any $u, v \in Q_\mathcal{A}^r$, $\delta(uav) = a$. It is said to be *quiescent* if $\delta(a^{2r+1}) = a$.

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

In addition, 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.

A σ-invariant 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

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

The simplest and most natural class of computable and σ-invariant measures is that of *Bernoulli measures*: they correspond to the case where each cell of a configuration is chosen independently according to a common fixed probability law over the alphabet.

Definition 2.3 (Bernoulli measure)**.** For an alphabet Q, a *Bernoulli measure* is a measure μ such that:

$$\forall u \in Q^*, \forall i \in \mathbb{Z}, \mu([u]_i) = \prod_{q \in Q} \mu([q]_0)^{|u|_q}.$$

The state probabilities $\mu([q]_0)$ are called the *coefficients* of μ. μ has full support if all coefficients are non-null.

The *uniform Bernoulli measure* μ_0 is the Bernoulli measure whose coefficients are all equal, equivalentely it is defined by:

$$\forall u \in Q^*, i \in \mathbb{Z}, \mu_0([u]_i) = \frac{1}{|Q|^{|u|}}$$

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

Definition 2.4 (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 weakly 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]. It is defined by their language as follows.

Definition 2.5 (Persistent set)**.** For a CA $\mathcal{A}$, define the *persistent set* $L_\mu(\mathcal{A}) \subseteq Q^*$ by: $\forall u \in Q^*$:

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

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 2.6 (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 2.7 (Set of predecessors)**.** Define the set of predecessors at time t of a finite word u for a CA $\mathcal{A}$ as $P^t_{\mathcal{A}}(u) = \left\{v \in Q^{|u|+2rt} : \mathcal{A}^t([v]_{-rt}) \subseteq [u]_0\right\}$.

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

Lemma 2.2. *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:*

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

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

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

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

This concludes the proof. □

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

Definition 2.8 (Cesaro-persistent set)**.** For a CA $\mathcal{A}$, we define the *Cesaro-persistent set* $C_\mu(\mathcal{A}) \subseteq Q^*$ by: $\forall u \in Q^*$:

$$u \notin C_\mu(\mathcal{A}) \iff \lim_{n\to\infty} \frac{1}{n}\sum_{k\leq n} \mathcal{A}^k\mu([u]_0) = 0.$$

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

We then get a lemma equivalent to Lemma 2.2 but for the Cesaro-persistent set. Its proof is the same.

Lemma 2.3. *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:*

$$u \notin C_\mu(\mathcal{A}) \iff \lim_{t\to+\infty} \frac{1}{t} \sum_{\tau=0}^{t} d_{\mathcal{A}^\tau(c)}(u) = 0$$

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 position 0 time t is the probability to have 0^{2t+1} centered on position 0 in 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 $\Lambda_\mu(\mathcal{A}_M) = \{^\omega 1^\omega\}$.

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 $\Lambda(\mathcal{A}_M) = (^\omega 10^* 1^\omega) \cup (^\omega 0^\omega) \cup (^\omega 10^\omega) \cup (^\omega 01^\omega)$. Actually, we can prove that this limit set is an example of limit set that cannot be a μ-limit set [BDS10].

Example 2.2. Consider any CA $\mathcal{A}$ over alphabet Q and add to it a spreading state $s \notin Q$: if a cell sees s in its neighborhood it becomes s, otherwise it behaves according to $\mathcal{A}$. By the same reasoning as above, the new CA $\mathcal{A}_s$ obtained this way has a trivial μ-limit set (as soon as μ gives some weight to s): the singleton made of configuration $^\omega s^\omega$. On the other hand, its limit set is as complex as the one from $\mathcal{A}$, precisely: its intersection with $Q^{\mathbb{Z}}$ is exactly the limit set of $\mathcal{A}$.

In [BPT06], it is shown that the μ-limit set of the elementary CA 184 is exactly the pair of configurations $\left\{^\omega(01)^\omega, ^\omega(10)^\omega\right\}$ when μ is the uniform measure. It is interresting to note that, on the contrary, a limit set must be a singleton when it is finite.

Other examples are studied in detail in [KM00].

3 Construction Toolbox

This section is dedicated to the proof of the following theorem.

Theorem 3.1. *Given a finite alphabet Q_0:*

1. *for any growing computable sequence $w \in \mathbb{W}(Q_0)$, there exists a CA $\mathcal{A}$ over alphabet $Q \supseteq Q_0$ such that, for any full-support Bernoulli measure μ over Q, $L_\mu(\mathcal{A}) = L_w$.*

2. *for any growing computable sequences $w, w' \in \mathbb{W}(Q_0)$, there exists a CA $\mathcal{A}$ over alphabet $Q \supseteq Q_0$ such that, for any full-support Bernoulli measure μ over Q, $\begin{cases} L_\mu(\mathcal{A}) = L_w \cup L_{w'} \\ C_\mu(\mathcal{A}) = L_{w'} \end{cases}$.*

It will mainly be used as a tool but it has an immediate corollary that gives an interesting hint on what is the set of all possible μ-limit sets (recall that we fixed a global set $\mathcal{Q}$ from which we take any finite alphabet, hence the set of cellular automata or the set of μ-limit sets is well-defined).

Corollary 3.2.

$$\begin{aligned}&\{L_\mu(\mathcal{A}), \mathcal{A} \textit{ is a CA and } \mu \textit{ the uniform measure }\}\\ =&\{C_\mu(\mathcal{A}), \mathcal{A} \textit{ is a CA and } \mu \textit{ the uniform measure}\}\\ =&\{L_w, w \in \mathbb{W}(Q) \textit{ and } Q \textit{ is a finite alphabet}\}\end{aligned}$$

Proof. Using Theorem 3.1, the only part that remains to be proven is that any $L_\mu(\mathcal{A})$ and any $C_\mu(\mathcal{A})$ is of the form L_w for some w. Indeed: consider a computable generic configuration c and define w_t as the word of size t at the center of $\mathcal{A}^t(c)$. The sequence $\left(d_{w_t}(u)\right)_t$ converges towards $\left(d_{\mathcal{A}^t(c)}(u)\right)_t$ therefore Lemma 2.2 concludes that $L_w = L_\mu(\mathcal{A})$ where $w = (w_t)$.

Now if we define w'_t as the concatenation of the words of size t at the center of $\mathcal{A}^i(c)$ for all $0 \leq i \leq t$ we get that the sequence $\left(d_{w'_t}(u)\right)_t$ converges towards the sequence

$$\left(\frac{1}{t}\sum_{i\leq t} d_{\mathcal{A}^i(c)}(u)\right)_t$$

Lemma 2.3 concludes that $L_{w'} = C_\mu(\mathcal{A})$ where $w' = (w'_t)$. □

Note that the use of the uniform measure is not essential in the above corollary. The same proof works for any Bernoulli measure with full support and computable coefficients.

The proof of Theorem 3.1 is constructive and consists in the description of the CA realizing the desired μ-limit set. This section will successively deal with the different parts of the construction after a short overview of the ideas we use.

3.1 Overview

We describe a CA $\mathcal{A}$ over alphabet Q that contains the alphabet Q_0 of the theorem. This CA has 3 components which work essentially independently but achieve together the desired behavior. The general idea is that, starting from a random configuration, $\mathcal{A}$ will:

- self-organizes into well-structured computation zones;
- these zones will evolve with time thanks to a merging process that ensures that they grow in size at a controlled rate;
- inside these well-sized zones a computation process runs permanently and essentially fills the zone in an appropriate way with the words from the given growing computable sequences.

We will describe $\mathcal{A}$ in a incremental way since each stage of the behavior above makes sense only within some structured zones prepared by the previous stage. However, all these stages actually run in parallel and at any time there are zones of the configuration which are completely out of control. The point is that we will do a reasoning *in density* starting from a generic configuration, which justified by lemmas 2.2 and 2.3.

More precisely, the incremental description and analysis of $\mathcal{A}$ will be the following:

- Cleaning out the space (Section 3.2, alphabet Q_1): *the density of reliable cells goes to 1.*
- Centralization (Section 3.3, alphabet Q_2 containing a quiescent state 0_2): *the density of cells belonging to a well-sized computation zone (size depending on time) goes to 1.*
- Computing and writing (Section 3.4, alphabet Q_3 containing a quiescent state 0_3): *within a well-sized computation zone, the content is filled with copies of w_i or w'_i (depending on time) up to some set of cells whose density goes to* 0.

The alphabet of $\mathcal{A}$ is $Q = Q_0 \sqcup ((Q_1 \sqcup (Q_2 \times Q_3)) \times Q_0)$. This can be interpreted the following way:

- every cell contains a layer filled with some state in Q_0, this is the primary layer;
- some cells contain additionnally another layer (secondary layer) containing either a state in Q_1 or in $Q_2 \times Q_3$.

3.2 Cleaning out the Space

In this section, we describe the initialization of the construction. Only the secondary layer is concerned and unless stated otherwise, all the states mentionned are in alphabet Q_1. 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.

3.2.1 General Description

The idea is to use a special state $\boxed{*} \in Q_1$ 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, apart from another counter, is erased. This is a key point, since we will only be interested in well formed structures, that is two walls moving away one from the other and delimiting a totally controled space. 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.

3.2.2 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 pair 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 smallest 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.

3.2.3 Dealing with collisions

Collisions of signals are handled in the following way:

- an outer signal carries a bit: it can be *open* or *closed*;
- nothing other than an outer signal can go through a *closed* outer signal (in particular, no "naked information" not contained between counters);
- when two *outer* signals collide they move through each other, both become *open*, and comparison signals are generated as illustrated by Figure 1:
  - on each side, a signal S1 moves at maximal speed towards the *inner* border of the counter, bounces on it (C and C') and goes back as S2 to the point of collision (D);

  - the first signal S2 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.

- nothing else than an outer signal or a S2 can go through an *open* outer signal;
- when a signal S2 goes through an *open* outer signal, this one becomes *closed*;
- nothing else than an inner signal or an outer signal can stop a signal S1; so an S1 signal, either encounters an inner signal and bounce on it and becomes S2, or it is destrotyed by another outer signal.

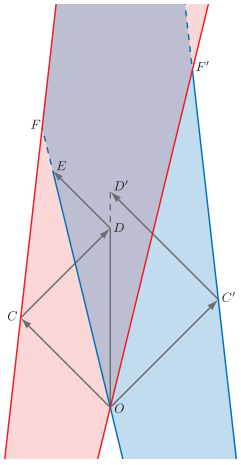

Figure 1: The bouncing signal must arrive (point E) before the older counter moves through the younger one (point F).

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 i, depending on the current state we would consider it to be exactly at the position i, or $(i+1/n)$, or $(i+2/n)$, 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 (it is the useful case for the other parts of the 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 $\boxed{\#} \in Q_2$ (this is the only state outside of Q_1 that we consider in this section) is then written and remains where the collision happened, as illustrated in Figure 2.

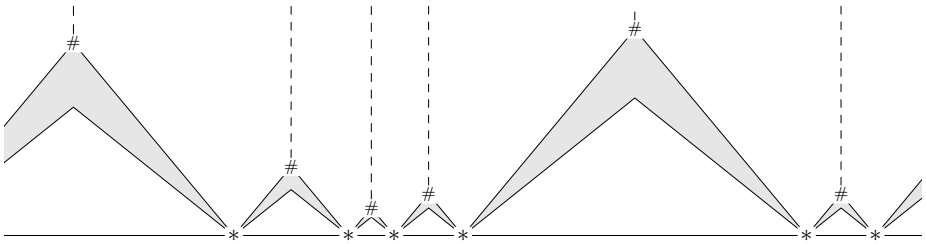

Figure 2: Collision of counters of same age.

Definition 3.1. Given any initial configuration c, a cell z at time t is *reliable* if it is inside the inner cone of some state $\boxed{*}$ in the initial configuration.

Note that this definition is independent of the parts of the CA which are not yet described. Therefore we can already prove a density result about reliable cells.

Lemma 3.3. *For any non-trivial Bernoulli measure μ and generic configuration c, the density of reliable cells goes to* 1 *as time increases.*

Proof. For a cell at time t and position z to be unreliable it is necessary that state $\boxed{*}$ does not occur in the initial configuration within the interval $[z - \lambda(t); z + \lambda(t)]$ (where λ is a linear function related to the slope of the inner cone). The configuration c being generic for a non-trivial Bernoulli measure, we immediately have that the density is upper bounded by $(1-p)^{2\lambda(t)}$ where $0 < p < 1$ is the probability of state $\boxed{*}$. The lemma follows. □

In the following we will focus on reliable parts of the configuration only.

3.3 Centralization

Definition 3.2. In a configuration c at time t, a segment is an interval $[z_1; z_2]$ of reliable cells not containing $\boxed{\#}$ and such that both $z_1 - 1$ and $z_2 + 1$ are in state $\boxed{\#}$.

In this section, we describe the external behavior of segments, that is how they interact. In the next section, we will need to dispose of arbitrarily large segments, and to get rid of the small ones. Thus, the idea is to erase some delimiters $\boxed{\#}$ in order to pool the available space of many segments into a single one. We will make sure that most segments eventually merge with another one, which means most segments become arbitrarily large through time. We still do not concern ourselves with the primary layer. For the secondary layer, all the states we will use in this section are in the alphabet $Q_2 \times Q_3$ that already contains $\{\boxed{\#}\} \times Q_3$. We will use the alphabet Q_3 later, hence suppose that the Q_3 component is always 0_3 (the quiescent state for Q_3) for the moment.

Now, let us describe the dynamics of segments among themselves. 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 an information shared by every segment. With such a protocol, mergings are many to one and not only two to one.

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 $K \geq 2$ (the value will be precised later), 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.

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

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 n in base K, then $\lfloor i + 1 \rfloor$ is computed inside the $\lceil \log_K(\sqrt{n}) \rceil$ leftmost cells. Now each segment knows its age and its size.

3.3.2 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 $i + 1$). If so, it checks whether its neighbors want to merge too. 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 only one among its neighbors wants to merge, it merges with that one.

Then each $\boxed{\#}$ delimiter between a segment and the segment it wants to merge with is erased. New segments are created between the remaining $\boxed{\#}$.

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 $i+1$ which takes time polynomial in $\sqrt{i}$ (value given by the length of the word encoding the age);
- compare both, linear time;
- check its neighbors: n timesteps (if we suppose they have achieved their own computations).

A segment needs to merge if $n \leq i + 1$, 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.

Remark 3.2.

1. For any $i \in \mathbb{N}$, after time t_i, each segment is larger than i.
2. If two segments exactly merge at time $t_i, i \in \mathbb{N}$, at least one of them is smaller than i.
3. If three or more segments merge together at time $t_i, i \in \mathbb{N}$, they are all smaller than i.

3.3.3 μ-limit sets

The purpose of this slow merging process is to control the size of segments so that the computation process we put inside can do its job correctly:

- we need larger and larger segments to do longer and longer computations,
- but but we want the typical size to grow slowly enough so that the computation process has time to fill the segment with the result of the computation.

We already saw that, by construction, segments at time t_i are of size at least i. So it remains to put an upper bound on the desired size of segments.

Definition 3.3. For all $i \in \mathbb{N}$, a segment is said to be *well-sized* at time $t_i \leq t < t_{i+1}$ if its size is greater than i and less than $K_i = i^3$.

The main goal of this section is to show that the density of cells that are inside a well-sized segment goes to 1. To show this we will focus on all cells that are in an undesirable situation, *i.e.* in one of the following cases:

1. unreliable,
2. reliable but not inside a segment.
3. inside a too large segment (too small segments can not exist by construction),

The first case was solved in the previous section (Lemma 3.3). We will now formalize the others cases. In the sequel, all the reasoning is done starting from a configuration c which is generic for some non-trivial Bernoulli measure. The general idea is to consider (at any time t) *maximal intervals* of cells which are reliable but not in state $\boxed{\#}$. For instance, a segment is such an interval with a $\boxed{\#}$ at both ends. To each interval we associate its corresponding pattern in the initial configuration. Then, to show that a certain kind of interval has a small density, it is sufficient to show that the corresponding pattern in the initial configuration has a sufficiently small probability compared to the length of the interval.

Lemma 3.4. *Let $d_0(t)$ (resp. $d_0(t,l)$) be the density of reliable cells at time t that are in a maximal reliable interval (resp. of size l) but not inside a segment. We have the following:*

1. $\exists \alpha, 0 < \alpha < 1$, *such that* $d_0(t,l) \leq \alpha^l$ *for any* t *and for* l *large enough;*
2. $d_0(t) \xrightarrow[t \to +\infty]{} 0$

Proof. Consider a maximal reliable interval $I \subseteq \mathbb{Z}$ which is not a segment at time t: at least one of its ends corresponds to the inner cone generated by some state $\boxed{*}$ at position $z_0 \in I$ in the initial configuration. So there is some constant $\alpha > 0$ (slope of the cone) such that $|I| \geq \alpha \cdot t$. Now, considering the history of I, i.e. the successive maximal reliable intervals $I_{t'}$ containing z_0 at any time t' between 0 and t, we have the following:

- $I_{t'}$ is never a segment;
- therefore, if a merging happens at some extremity of some $I_{t'}$, it can only be when $t' = t_i$ (for some i) and involve $I_{t'}$ plus a single segment of size at most i.

We deduce that $|I| \leq 2 \cdot \alpha \cdot t$ (the worst case being when I has not been involved in any merging in all its history). Therefore, there must be an interval $I_0 \subset I$ of cells which are not in state $\boxed{*}$ in the initial configuration and whose size verifies: $|I_0| \geq |I|/2$.

Denoting by $d_L(t,l)$ the density of cells which are the leftmost cell of a maximal reliable interval of size l which is not a segment at time t, we have:

1. $d_L(t,l) = 0$ if $l \leq \alpha \cdot t$;
2. $d_L(t,l) \leq l \cdot (1-p)^{l/2}$ otherwise,

where p is the probability of state $\boxed{*}$. Hence $d_0(t,l) \leq l^2 \cdot (1-p)^{l/2}$ for any t and the first item is proved. Moreover we deduce that the density $d_0(t)$ of reliable cells not inside a segment at time t verifies:

$$\begin{aligned} d_0(t) &\leq \sum_{l=\alpha t}^{+\infty} l^2 \cdot (1-p)^{l/2} \\ &\leq \sum_{l=\alpha t}^{+\infty} \left((1-p)^{\frac{1}{4}}\right)^l \text{ for } t \text{ large enough} \\ &= \frac{(1-p)^{\alpha t/4}}{1-(1-p)^{1/4}} \xrightarrow[t\to+\infty]{} 0. \end{aligned}$$

□

Lemma 3.5. *The density $d_1(t)$ of cells which are at time t in a segment which is not well-sized goes to 0 as $t \to \infty$.*

Proof. Let $d_+(i)$ be the density of cells at time t_i which are in a segment larger than K_i and consider a time t with $t_i \leq t \leq t_{i+1}$. Too small segments (*i.e.* smaller than i) can not exist by construction and too large segments can only come from too large segments at time t_i or reliable intervals which turned into segments at time t. So we have:

$$d_1(t) \leq d_+(i) + d_0(t-1).$$

Therefore, using Lemma 3.4, it is sufficient to prove that $d_+(i)$ goes to 0 when $i \to \infty$.

Consider a segment of size $l \geq K_i$ at time t_i. It may only come from one of the following situations:

1. a segment of size l at time t_{i-1},
2. the merging of a segment of size $l-i$ and another of size i at time t_{i-1},
3. a reliable interval which is not a segment of size at least $l-i$ at some time t, $t_{i-1} \leq t \leq t_i$,
4. the merging of many segments of size i at time t_i.

Going back in time recursively through cases 1 or 2 until encountering case 3 or 4, we deduce that to each segment of size $l \geq K_i$ at time t_i corresponds either a reliable interval which is not a segment of size at least $l - i^2$ at some time $t \leq t_i$, or a merging of many segments of size j at time t_j (for some $j \leq i$) resulting in a large segment of size at least $l - i^2$ (in both cases the reduction of i^2 in size is an upper bound on the worst case where we lose i at time t_i, $i-1$ at time

t_{i-1}, etc). Now, denoting by $d_S(i,l)$ the density of cells which are in a segment of size l at time t_i, and $d_M(j,k)$ the density of cells which are at time t_j in a segment of size k resulting from a merging of many segments of size j, we have:

$$d_S(i,l) \leq \frac{l}{l-i^2}\big(\sum_{j\leq i} d_M(j,l-i^2) + \sum_{t\leq t_i} d_0(t,l-i^2)\big) \tag{1}$$

Let's focus first on $d_M(j,k)$ and consider a segment at time t_j coming from a merge of k/j segments of size j. Let's $z_0 \in \mathbb{Z}$ be the leftmost position of that segment: by hypothesis, this implies that at each position $z_0 + nj$, for $0 \leq n \leq k/j$, a $\boxed{\#}$ is created at some time before t_j. This in particular implies that the specific pattern $P = \boxed{*}\,a\,\boxed{*}\,a\,\boxed{*}$ (where $a \neq \boxed{*}$) does not occur centered at any of the aforementioned positions $z_0 + nj$ in the initial configuration: indeed, it would create a $\boxed{\#}$ at positions $z_0 + nj - 1$ and $z_0 + nj + 1$ at time 1, and forbid forever the apparition of the required $\boxed{\#}$ at position $z_0 + nj$. Let q be the probability of this pattern P, we deduce that the density of such positions as z_0 is less than

$$(1-q)^{k/j}$$

and therefore

$$d_M(j,k) \leq k \cdot (1-q)^{k/j}.$$

Now putting back this upper bound on $d_M(j,k)$ and the one from Lemma 3.4 on $d_0(t,l)$ into Equation 1, we get

$$d_S(i,l) \leq \frac{l}{l-i^2}\big(i \cdot l \cdot \beta^{\frac{l-i^2}{i}} + t_i \cdot \alpha^{l-i^2}\big)$$

where α and β are constants between 0 and 1. We therefore have

$$d_+(i) \leq \sum_{l\geq K_i} \frac{l}{l-i^2}\big(i \cdot l \cdot \beta^{\frac{l-i^2}{i}} + t_i \cdot \alpha^{l-i^2}\big)$$

Since t_i grows like $K^{\sqrt{i}}$ and K_i grows like i^3, there is some constant γ between 0 and 1 such that, for large enough i, each term of the sum above is less than γ^l (for all $l \geq K_i$). The lemma follows because for large enough i we then have:

$$d_+(i) \leq \sum_{l\geq K_i} \gamma^l = \frac{\gamma^{K_i}}{1-\gamma} \xrightarrow[i\to+\infty]{} 0.$$

□

From the two lemmas above, we deduce the main result of this section.

Proposition 3.6. *The density of cells which are inside a well-sized segment at time t goes to 1 as $t \to \infty$.*

Proof. The density of such cells is exactly $1 - d_{NR}(t) - d_0(t) - d_1(t)$ where $d_{NR}(t)$ denotes the density of non-reliable cells at time t. Lemmas 3.3, 3.4 and 3.5 concludes the proof. □

3.4 Computing and Writing

In this section we describe the final part of the construction. We concentrate on the internal behavior of the segments, and we will use the states of the alphabet $Q_2 \times Q_3$. The Q_2 part was described in the previous section, and the Q_3 part does not interfere with it, hence the state is in $\{O_2\} \times Q_3$ when the Q_2 part is not specified by the rules of the previous section.

The computation inside segments depends on two growing sequences $(w_i)_i$ and $(w'_i)_i$ which can each be generated on input i in time $\frac{1}{4}(t_{i+1} - t_i)$ and space $\sqrt{i}$.

In a segment at time $t_i, i \in \mathbb{N}$, the computation process goes through the following steps (see figure 3):

- **(a)** in the $\lfloor\sqrt{i}\rfloor$ leftmost cells, the Turing machines T and T' compute and output the words w_i and w'_i;
- **(b)** a writing head carrying a memory writes copies of the word w_i separated by some delimiter $\boxed{€}$;
- **(c)** the writing head comes back to the left of the segment and wait until $t = t_i + \frac{1}{2}(t_{i+1} - t_i) + (|w_i| + 1)K_i$;
- **(d)** a writing head carrying a memory writes copies of the word w'_i separated by some delimiter $\boxed{€}$, thus erasing the copies of w_i;
- **(e)** the writing head kills itself.

Suppose the length of the segment is l, recall $l \geq i$.

For the first part **(a)**, the Turing machines are simulated successively in the obvious way, and thanks to the previous remark, they never use more than $\lfloor\sqrt{i}\rfloor$ cells, and the computation is achieved before time $t_i + \frac{1}{2}(t_{i+1} - t_i)$.

For the second part **(b)**, once the words w_i and w'_i computed and stored in the $\lfloor\sqrt{i}\rfloor$ leftmost cells of the segment, a prefix of the word $(w_i\boxed{€})^\omega$ is written all over the segment, this is achieved with a head that carries w_i as its memory, hence it needs $|w_i|.l$ timesteps to reach the end of the segment.

The third part **(c)** takes only l steps, thus, for any well-sized segment, the head arrives at the left of the segment before time $t_i + \frac{1}{2}(t_{i+1} - t_i) + (|w_i| + 1)K_i$. In the case of a non well-sized segment, the writing process stops there.

The fourth part **(d)** is similar to the third one and the fifth part **(e)** is instantaneous.

The whole process takes less than $\frac{1}{4}(t_{i+1} - t_i) + \frac{1}{4}(t_{i+1} - t_i) + (|w_i| + 1)K_i + |w'_i|l \leq \frac{1}{2}(t_{i+1} - t_i) + (2\sqrt{i} + 1)l$ which is less than $t_{i+1} - t_i$ for any well-sized segment ($l \leq K_i$).

For any segment s at time t, denote $w_t(s)$ the content of s, that is $F^t(c)_{[a,b]}$ if the segment is between positions a and b. Therefore, for any s and t, $w_t(s)$ 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.

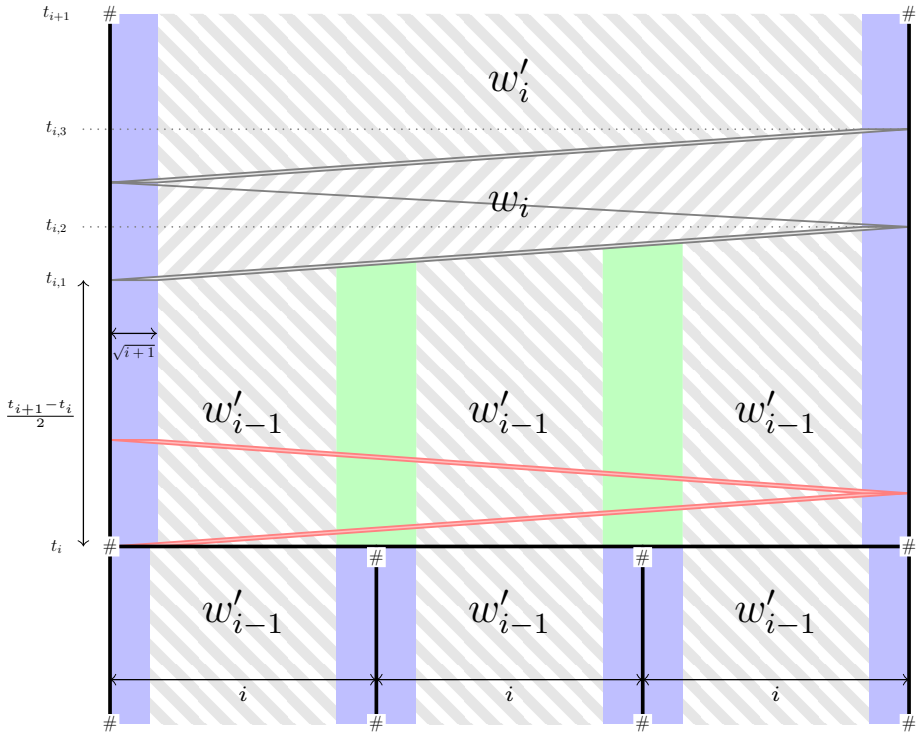

Figure 3: Computation process in a well-sized segment after a merging.

3.5 Proof of the theorem

Recall the statement of the theorem:

Theorem 3.7. *Given a finite alphabet Q_0:*

1. *for any growing computable sequence $w \in \mathbb{W}(Q_0)$, there exists a CA $\mathcal{A}$ over alphabet $Q \supseteq Q_0$ such that $L_\mu(\mathcal{A}) = L_w$ where μ is a full-support Bernoulli measure over Q.*

2. *for any growing computable sequences $w, w' \in \mathbb{W}(Q_0)$, there exists a CA $\mathcal{A}$ over alphabet $Q \supseteq Q_0$ such that $\begin{cases} L_\mu(\mathcal{A}) = L_w \cup L_{w'} \\ C_\mu(\mathcal{A}) = L_{w'} \end{cases}$ where μ is a full-support Bernoulli measure over Q.*

We will prove the second part of the theorem. We then deduce the first point by taking $w_i = w'_i, \forall i \in \mathbb{N}$. First, by lemma 2.1, remark that it is possible to suppose that the growing computable sequences $(w_i)_i$ and $(w'_i)_i$ are such that there exist Turing machines T and T' such that, given $i \in \mathbb{N}$ as an input, T (resp. T') computes w_i (resp. w'_i) in time $\frac{1}{4}(t_{i+1} - t_i)$ and space $\sqrt{i}$.

The idea of the proof is that well-sized segments that result from a merging of well-sized segments, called *good segments*, tend to almost cover the images of

a generic configuration (direct corollary of Proposition 3.6), and they contain essentially copies of the words w_i or w'_i. The technical point justifying to focus only on good segments and not on all well-sized segment is that a well-sized segment that just merged from a not well-sized segment might not be properly initialized.

It is essential to note that the content of good segments is easily described as said in the following remark.

Remark 3.3. For $t_i \leq t \leq t_{i+1}$ and any good segment s, $w_t(s) = v_0v_1v_2v_3v_4$ where:

- $|v_0| \leq \sqrt{i}$, this corresponds to the computation area and the storage of the age counter;
- $|v_4| \leq \sqrt{i}$, this corresponds to the storage of the age counter on the right;
- $|v_2| \leq \sqrt{i}$, this corresponds either to the signal that computes $|s|$ ($t \leq t_i + \frac{1}{2}(t_{i+1} - t_i)$) or to the writing head and its memory ($t \geq t_i + \frac{1}{2}(t_{i+1} - t_i)$);
- v_1 and v_3 belong to Q_0^*.

The words v_2 and v_3 may be empty. Moreover, v_1 and v_3 contain periodic repetitions of w'_{i-1} (written before t_i), w_i or w'_i, depending of the status of the writing process. In particular, for $t = t_i$, v_2 and v_3 are empty and v_1 contains repetitions of w'_{i-1}. At $t = t_{i+1} - 1$, v_2 and v_3 are empty and v_1 contains repetitions of w'_i.

Denote for $i \in \mathbb{N}$:

- $t_{i,1} = t_i + \frac{1}{2}(t_{i+1} - t_i)$: at that time, the writing process starts in good segments;
- $t_{i,2} = t_i + \frac{1}{2}(t_{i+1} - t_i) + (|w_i|)K_i$: at that time, copies of w_i€ are written all over well-sized segments;
- $t_{i,3} = t_i + \frac{1}{2}(t_{i+1} - t_i) + (|w_i| + |w'_i| + 1)K_i$: at that time, the writing process is finished.

First, the following Lemma justifies that we focus on the density of words inside good segments.

Lemma 3.8. *Take $u \in Q^*$ and c a generic configuration, and consider (α_t) such that $d_{w_t(s_t)(u)} \geq \alpha_t$ for any good segment s_t at time t starting from c. We have the following:*

1. *if $\alpha_t \not\to 0$ then $u \in L_\mu(\mathcal{A})$*
2. *if α_t does not go to 0 in Cesaro mean then $u \in C_\mu(\mathcal{A})$*

Proof. From Proposition 3.6, for any $\varepsilon > 0$ and for any large enough t, the density of cells outside good segments is at most ε (because the density at step

t_{i+1} of well-sized segments which are not good is less than the density at step t_i of segments which are not well-sized). Therefore we have from the hypothesis:

$$d_{\mathcal{A}^t(c)}(u) \geq (1-\varepsilon)\cdot\alpha_t$$

Lemma 2.2 and 2.3 then allow to conclude. □

Claim 3.9. $L_{w'} \subseteq C_\mu(\mathcal{A})$

Proof. Let $u \in L_{w'}$ and c a generic configuration.

There exists $\varepsilon > 0$ such that $\forall i_0 \in \mathbb{N}, \exists i \geq i_0$ such that $d_{w'_i}(u) > \varepsilon$. We will consider $t \in \mathbb{N}$ such that $t_{i,3} \leq t \leq t_{i+1}-1$. For i large enough, in every good segment s_t at time t, v_0 and v_4 are negligible in the description of Remark 3.3. Therefore

$$d_{w_t(s_t)}(u) \geq \frac{1}{2} d_{(w'_i\boxed{\text{€}})^\omega}(u) \geq \frac{1}{4} d_{w'_i}(u) \geq \frac{1}{4}\varepsilon.$$

So we put $\alpha_t = \frac{1}{4}\varepsilon$ for t verifying $t_{i,3} \leq t \leq t_{i+1}-1$ (i large enough) and $\alpha_t = 0$ elsewhere. Summing at time $t_{i+1}-1$, we get:

$$\begin{aligned}
\frac{1}{t_{i+1}-1}\sum_{t=0}^{t_{i+1}-1}\alpha_t \geq & \frac{1}{t_{i+1}-1}\sum_{t=t_{i,3}}^{t_{i+1}-1}\alpha_t \\
\geq & \frac{1}{t_{i+1}-1}(t_{i+1}-1-t_{i,3})\frac{1}{4}\varepsilon \\
\geq & \frac{1}{16}\varepsilon
\end{aligned}$$

We conclude thanks to Lemma 3.8. □

Claim 3.10. $L_w \subseteq L_\mu(\mathcal{A})$

Proof. Similarly as in the previous proof, there exists $\varepsilon > 0$ such that for arbitrarily large i $d_{w_i}(u) > \varepsilon$. Then we can lower-bound the density of u in any good segment at time $t_{i,2}$ by $\frac{1}{4}\varepsilon$, and we conclude thanks to Lemma 3.8. □

Claim 3.11. $L_\mu(\mathcal{A}) \subseteq L_{w'} \cup L_w$

Proof. Take $u \notin L_{w'} \cup L_w$, for any $t \in \mathbb{N}$, the density of u in $\mathcal{A}^t(c)$ is due to occurences of u outside good segments (density $d_t(u)$) and occurences inside good segments (density $d'_t(u)$). Since the density of cells outside good segments tends to 0 (Proposition 3.6), $d_t(u) \to_{t\to\infty} 0$.

Inside good segments, the density of u is less than the sum of densities of u in w_i, in w'_i and in cells corresponding to v_0, v_2 or v_4 in the description of these segments made in Remark 3.3, that is $\forall t_i \leq t < t_{i+1}, d'_t(u) \leq d_{w_i}(u) + d_{w'_i}(u) + \varepsilon_i$ with $\varepsilon_i \to_{i\to\infty} 0$. This proves the claim, using Lemma 2.2. □

Claim 3.12. $C_\mu(\mathcal{A}) \subseteq L_{w'}$

Proof. As $C_\mu(\mathcal{A}) \subseteq L_\mu(\mathcal{A})$, it is enough to prove that $\forall u \in L_w \setminus L_{w'}, u \notin C_\mu(\mathcal{A})$. Take such an u and a generic configuration c.

For $i \in \mathbb{N}$, recall $t_{i,1} = \frac{1}{2}(t_{i+1} - t_i)$ and $t_{i,3} = \frac{1}{2}(t_{i+1} - t_i) + (|w_i| + |w'_i| + 1)K_i$. For any $t \in \mathbb{N}$ such that $t_i \leq t \leq t_{i,1}$ or $t_{i,3} \leq t < t_{i+1}$, it is possible to bound $d_{\mathcal{A}^t(c)}(u)$ by some ε_i with $\varepsilon_i \to_{i\to\infty} 0$. Indeed, for such t, in every good segment, in the description of Remark 3.3, v_1 and v_3 contain copies of w'_i or w'_{i-1}; and cells outside good segment have a density going to zero (from Proposition 3.6). To simplify the proof, let's choose ε_i such that it is also a bound on the density of cells outside good segments for any t between t_i and t_{i+1}.

Then for every good segment at time t_i:

$$\sum_{t=t_i}^{t_{i+1}-1} d_{w_t s}(u) = \sum_{t=t_i}^{t_{i,1}} d_{w_t s}(u) + \sum_{t=t_{i,1}}^{t_{i,3}} d_{w_t s}(u) + \sum_{t=t_{i,3}}^{t_{i+1}-1} d_{w_t s}(u)$$

We deduce that:

$$\begin{aligned}
\sum_{t=t_i}^{t_{i+1}-1} d_{\mathcal{A}^t(c)}(u) &\leq \tfrac{1}{2}(t_{i+1} - t_i)\varepsilon_i + (|w_i| + |w'_i| + 1)K_i + \tfrac{1}{2}(t_{i+1} - t_i)\varepsilon_i + (t_{i+1} - t_i)\varepsilon_i \\
\sum_{t=t_i}^{t_{i+1}-1} d_{\mathcal{A}^t(c)}(u) &\leq 2(t_{i+1} - t_i)\varepsilon_i + (2\sqrt{i} + 1)K_i
\end{aligned}$$

Now take $t \in \mathbb{N}$, there exists $i \in \mathbb{N}$ such that $t_i \leq t < t_{i+1}$, hence

$$\begin{aligned}
\tfrac{1}{t}\sum_{\tau=0}^{t} d_{\mathcal{A}^\tau(c)}(u) &\leq \tfrac{1}{t}\sum_{j=0}^{i}\sum_{\tau=t_j}^{t_{j+1}-1} d_{\mathcal{A}^\tau(c)}(u) \\
&\leq \tfrac{1}{t}\sum_{j=0}^{i}\left(2(t_{j+1} - t_j)\varepsilon_j + (2\sqrt{j} + 1)K_j\right) \to_{t\to\infty} 0
\end{aligned}$$

□

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

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

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 from 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 for every fully supported Bernoulli measure μ.*

Proof. We already know that this problem is Σ_3 at most. We will use Theorem 3.1 to prove the completeness. Let us describe the growing computable sequence $(w_i)_{i\in\mathbb{N}}$ that will be used.

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 $f(i) = (j,k,l)$. (Any such enumeration will do.) Let $(\phi_n)_{n\in\mathbb{N}}$ be a computable enumeration of Turing machines. We describe the Turing machine T such that T outputs w_i when given the input i.

Take $i \in \mathbb{N}$, there are $j,k,l \in \mathbb{N}$ such that $f(i) = (j,k,l)$. The idea is to simulate the computation of the Turing machine ϕ_j on some particular sequence of consecutive inputs then choose w_i according to the results of the computations, i.e. depending whether the machine halts on each input in this sequence or not. We will say that i 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 the computation ends on all sequences $\{k, k+1, \ldots, k+l\}$. Thus, w_i will be a witness (taking the form of a prefix of $(\boxed{\$_0}\square^j\boxed{\$_1}\square^k\boxed{\$_2})^\omega$) of the success of a sequence starting at k. We will have to avoid writing a witness for k more than once for each l.

More formally, T does the following on input i:

- Compute $(j,k,l) = f(i)$.
- 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 sequence of simulations after 2^i steps (the bound is purely arbitrary). In this case, i is said to be failed.
- If the machine ϕ_j does halt on all these inputs before timestep 2^i, then denote τ the exact time used for the whole computation.
- If $\tau \leq 2^{i_0}$, then i is said to be failed again, since in this case, some smaller integer was declared successful with the same sequence.
- In the remaining case, i is said to be successful and $w_i = (\boxed{\$_0}\square^j\boxed{\$_1}\square^k\boxed{\$_2})^i$.
- If i is failed, $w_i = \square^{i(j+k+3)}$.

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

$$u_j \in L_w \Leftrightarrow \phi_j \in COF$$

Claim 4.5. $\phi_j \in COF \Rightarrow u_j \in L_w$

Proof of the claim:

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 2^i timesteps are enough to simulate ϕ_j on inputs $k, k+1, \ldots, k+l$ and verify that it halts in each case. Thus for every triplet $(j,k,l), l \in \mathbb{N}$, there exists a successful $i_l \in \mathbb{N}$ with $f(i_l) = (j,k,l)$.

Hence, $w_{i_l} = (\boxed{\$_0}\square^j\boxed{\$_1}\square^k\boxed{\$_2})^{i_l}$. For l, and thus i_l, large enough, the density of the word u_j in every w_{i_l} is larger than $\frac{1}{2}\frac{1}{|j|+|k|}$ which is a constant.

■

Claim 4.6. $\phi_j \notin COF \Rightarrow u_j \notin L_w$

Proof of the claim: Here, with j fixed, if we take an infinite number of successful integers, they necessarily concern unbounded values of the starting point k. We will use the fact that the density of u_j in $(\boxed{\$_0}\square^j\boxed{\$_1}\square^k\boxed{\$_2})^\omega$ 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 $d_{w_i}(u_j) = 0$.

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 $(j,k,l), k \leq k_0, l \leq l_0$:

- either every $i \in \mathbb{N}$ with $f(i) = (j,k,l)$ is failed;
- or there exists $i < i_0$ such that i is successful.

Now take $i \geq i_0$.

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

■
□

4.3 Descriptive complexity

In this section we will use Theorem 3.1 to construct cellular automata whose μ-limit sets are constrained to be in a specific subshift. The following proposition shows that we can build a μ-limit inside any effective subshift. However, let's recall that there are very simple effective subshits which can not be the μ-limit set of some CA as shown in Example 2.1. The question of what kind of subshift can appear as μ-limit sets has been specifically adressed in [BDS10].

Proposition 4.7. *Given a non-empty effective subshift S over an alphabet Q_0, there exists a CA whose μ-limit set is included in S for every fully supported Bernoulli measure μ.*

Proof. Because the subshift S is effective, it can be characterized by a recursively enumerable set of forbidden words. We will use Theorem 3.1 with a growing computable sequence in which the word w_i does not contain any of the first i forbidden words.

Let us describe the Turing machine T that computes the sequence $(w_i)_i$.

- On input i, T enumerates and stores the first i forbidden words of S.
- All possible words of length i over Q_0 are then enumerated in lexicographical order, and w_i is the first one that does not contain any of the forbidden words previously enumerated (there exists one because the subshift is non-empty).

We now apply Theorem 3.1 with the growing computable sequence $(w_i)_i$ hence it is enough to prove that L_w contains only words in Σ^* and none of the forbidden words.

Now let us consider a forbidden word v in the recursively enumerable set that characterizes the subshift S. It is the i^{th} word enumerated for some $i \in \mathbb{N}$, hence it does not appear in $w_j, j \geq i$. □

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 for every fully supported Bernoulli measure μ.*

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 $x \in \{0,1\}^*$ such that $K_f(x)$ is defined, 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 a shortest program which outputs that word.

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 for every fully supported Bernoulli measure μ.*

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 for every fully supported Bernoulli measure μ.*

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 for every fully supported Bernoulli measure μ.*

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. Recall that we have since the beginning consider a countable set $\mathcal{Q} = \{q_0, q_1, \dots\}$ from which we take finite subsets to define alphabets.

Definition 5.1. A property $\mathcal{P}$ of μ-limit sets of cellular automata is a subset of the powerset $\mathscr{P}(\mathcal{Q}^{\mathbb{Z}})$. 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 $\{q_i^{\mathbb{Z}}, i \in \mathbb{N}\}$. 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. 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 computable weakly generic configuration c_{WG} 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:*

$$A|X|^{-l} \leq d_{c_{WG}[0,L-1]}(u) \leq B|X|^{-l}$$

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

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

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. *There is an algorithm that, given a cellular automaton $\mathcal{A}$ and a Turing machine ϕ, produces a cellular automaton $\mathcal{B}$ such that:*

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

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

Proof of Theorem 5.2. 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. We reduce the decision problem $\mathcal{P}$ to COF as follows. 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 the oracle for $\mathcal{P}$ 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 $\phi \in COF$. So we decided COF on ϕ. □

Now we prove Proposition 5.3.

Proof of Proposition 5.3. The proof will have similarities with the one of Theorem 4.4. Denote for this proof $c = c_{WG}$.

It mainly relies on Theorem 3.1. Again, we make a reduction to the problem of being co-finite for a Turing machine which is Σ_3^0-complete. Let us describe the computable sequence $w = (w_i)_{i\in\mathbb{N}}$ associated to it. 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 $f(i) = (k, l)$. Denote T the Turing machine that produces w.

For $(k, l) = f(i), i \in \mathbb{N}$, the idea is to simulate the computation of ϕ on some sequence of consecutive inputs $(\{k, k + 1, \dots, k + l\})$ and output different w_i's 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 and failed in the other case. 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 witness of the success of a sequence starting at k. We will have to avoid writing a witness for k more than once for each l.

More formally, T does the following on input i:

- Compute $(k, l) = f(i)$ and $\nu(i) = \lfloor \log i \rfloor$.
- Compute $i_0 = \max\{i' < i, f(i') = (k, l)\}$.
- 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 2^i steps. In this case, i is said to be failed.
- If the machine ϕ does halt on all these inputs before timestep 2^i, then denote τ the exact time used for the whole computation.
- If $\tau \leq 2^{i_0}$, the whole computation necessarily ended for i_0 and again i is said to be failed.
- Compute $u_i = c_{[o..(\nu(i)-1)]}$ and $v_i = \left(\mathcal{A}^{\nu(i)}(c)\right)_{[o..(\nu(i)-1)]}$.
- If i is failed, define $w_i = v_i^{k+1}$.
- In the other case, i is said to be successful: define $w_i = u_i v_i^k$.

Claim 5.4. *If $\phi \in COF$ then $L_w = Q_{\mathcal{A}}^{\mathbb{Z}}$.*

Proof of the claim:

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 $l \in \mathbb{N}$, there exists $i_l \in \mathbb{N}$ such that $f(i_l) = (k, l)$ and 2^{i_l} timesteps are enough to simulate ϕ on inputs $k, k+1, \ldots, k+l$ and verify that it halts in each case. Thus there are infinitely many successful i's with $f(i) = (k, l)$ for some $l \in \mathbb{N}$.

Take any word $u \in Q_{\mathcal{A}}^*$. For any such large enough successful i ($\nu(i) \geq |Q_{\mathcal{A}}|^{2|u|}$), we hence have $d_{w_i}(u) \geq \frac{A}{2(k+1)|Q_{\mathcal{A}}|^{|u|}}$ (with A from Proposition 5.1) which is a constant as k is fixed.

■

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

Proof of the claim: 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 decreases when k increases.

Take $u \notin L_\mu(\mathcal{A})$ and $\varepsilon > 0$.

There exists $k_0 \in \mathbb{N}$, such that $\frac{1}{k_0+1} < \varepsilon/2$. 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 ${l_0}^2$ pairs $(k, l) \in \mathbb{N}^2$ with $k \leq k_0$ and $k + l \leq l_0$. Denote i_0 the smallest integer such that for any $f(i) = (k, l)$ with $i \geq i_0$ and $k \leq k_0$:

- either ϕ does not halt on some input between k and $k + l$;
- or there exists $i' \leq i_0$ with $f(i') = (k, l)$ such that ϕ halts on all these inputs in less than $2^{i'}$ timesteps.

Thus, when $f(i) = (k, l)$ with $i \geq i_0$ and $k \leq k_0$, i is failed and $w_i = \left(\left(\mathcal{A}^{\nu(i)}(c)\right)_{[o..(\nu(i)-1)]}\right)^{k+1}$.

Take $i_1 \geq i_0$ such that $\forall i \geq i_1, d_{\mathcal{A}^{\nu(i)}(c)}(u) < \varepsilon/(4B)$, which exists since $u \notin L_\mu(\mathcal{A})$.

Now take $i \geq i_1$ with $f(i) = (k, l) \in \mathbb{N}^2$.

- If $k \geq k_0$ then $d_{w_i}(u) < \frac{1}{k_0+1} + d_{\mathcal{A}^{\nu(i)}(c)_{[0..\nu(i)]}}(u) < \varepsilon$.
- If $k \leq k_0$, then i is failed and $d_{w_i}(u) < d_{\mathcal{A}^{\nu(i)}(c)_{[0..\nu(i)]}}(u) < \varepsilon$.

Hence, thanks to Remark 5.1, we conclude that $L_w \subseteq L_\mu(\mathcal{A})$.

The other direction is easier: for $u \in L_\mu(\mathcal{A})$ and $i \in \mathbb{N}$, $d_{w_i}(u) \geq \frac{1}{2} d_{\mathcal{A}^{\nu(i)}(c)_{[}0..\nu(i)]}(u)$ which does not tend to 0.

■

□

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 arbitrarily 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 5.6. *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 states 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 5.7. *Let μ be some Bernoulli measure on the fullshift, the problem of being μ-nilpotent for a cellular automaton is Π_3^0-complete.*

Proposition 5.8. *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. *For any fully supported Bernoulli measure μ, there exist F such that F and F^2 do not have the same μ-limit set.*

Proof. To construct such an F it is sufficient to use the counter construction from the proof of Theorem 3.1, *i.e.* the initialization step. 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 the proof of Theorem 3.1:

- 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 3.3, we can prove the following:

> **Claim.** Starting from a generic configuration, the density of cells which are synchronized at time t goes to 1 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 assumption simplifies the μ-limit set as well as some decision problems on it (to be compared to Theorems 4.4 and 5.7).

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 Π_3-hard when μ is the uniform Bernoulli 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 appearance 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 $\boxed{\$}$). First note

that in Theorem 3.1 the simple convergence of the CA is equivalent to the simple convergence of the densities of words in the computable sequence $(w_i)_i$. 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 $\liminf_i d_{w_i}([\$]) = 1$. Indeed, you get the result with a sequence $(i_j)_j$ where $f(i_j) = (0, 0)$. Hence, as in this case $L_w = Q_{\mathcal{A}}^*$ (Claim 5.4), the convergence cannot be simple.

□

Complexity considerations allow 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_3^0-hard μ-limit set. However, Theorem 6.2 shows that simply convergent CA produce μ-limit sets which are only Σ_2^0. □

6.2 Cesaro mean

The construction from Theorem 3.1 allowed us to build complex μ-limit sets. It also shows that this complexity can be completely wiped out when considering the Cesaro mean.

Theorem 6.4. *For any cellular automaton, there exists another one with the same μ-limit set (possibly up to a single uniform configuration) but which is μ-Cesaro nilpotent.*

Proof. It's a direct application of Theorem 3.1 where w is chosen so that L_w is the μ-limit set of the given CA and w' chosen so that $L_{w'}$ contains only 1 letter. □

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

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

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

Proof. To prove it we use again Theorem 3.1 with some growing computable sequence $w \in \mathbb{W}(Q)(0 \in Q)$ that we will describe. Note first that, due to the

construction used in the proof of Theorem 3.1, the first point of the theorem is implied by

$$\forall n \in \mathbb{N}, \exists i_n \geq \max\{T_m : m \leq n\}, d_{w_{i_n}}(0) \geq \frac{1}{n}$$

Take an enumeration f of the integers such that the preimage of any integer is infinite and $\forall i \in \mathbb{N}, f(i) \geq i$. For $i \in \mathbb{N}$ with $f(i) = m$, simulate the computation of machine m with input 0 during i steps. As in the proofs of Theorem 4.4 and Proposition 5.3, for each $m \in \mathbb{N}$, the smallest i such that the computation reaches its end is said to be successful and failed in the other case. If $i \in f^{-1}(m)$ is successful, take $w_i = (1^{m-1}0)^i$, else $w_i = 1^{im}$.

Thus 0 has density $\frac{1}{m}$ in the writing layers of segments only when the simulation of the machine halts for the smallest $i \in f^{-1}(m)$, and 0 otherwise. The two points of the result are now easily verified.

For the first point, given $n \in \mathbb{N}$, consider m_n such that $T_{m_n} = \max\{T_m : m \leq n\}$. There exists $i_n \in f^{-1}(m_n)$ successful, hence $d_{w_{i_n}}(0) \geq \frac{1}{n}$.

For the second point, given $n \in \mathbb{N}$, there exists $i_n \in \mathbb{N}$ such that every machine $m \leq n$ halts in less than i_n steps or never. Now, take $j_n \geq i_n$ such that every $m \leq n$ has been enumerated between i_n and j_n. Thus, for every $i \geq j_n$ the density of the word 0 is $d_{w_i}(0) \leq \frac{1}{n}$.

□

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_1^0-complete (see [Kar92])	Π_3^0-complete (see Thm. 5.7)
Any non-trivial property	Σ_1^0-hard (see [Kar94])	Π_3^0-hard (see Thm. 5.2)
Worst-case language	Π_1^0-complete (see [Hur87])	Σ_3^0-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 hypotheses on the input CA.

Type of input CA	**Worst L_μ**	**μ-Nilpotency**
General case	Σ_3^0-complete (see Thm. 4.4)	Π_3^0-complete (see Thm. 5.7)
Equicontinuous	Σ_1^0 (see Thm. 4.2)	Σ_2^0 (see Prop. 5.8)
Simply convergent	Σ_2^0 (see Thm. 6.2)	Π_2^0 (see Thm. 6.2)

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 full shift, 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.

Acknowledgment

We are grateful for the time spent by the anonymous referees on the first version of this paper and for the incitative to write a better version through their numerous comments.

References

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

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

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

[BPT06] L. Boyer, V. Poupet, and G. Theyssier. On the complexity of limit sets of cellular automata associated with probability measures. In *MFCS*, pages 190–201, 2006.

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

[dMS13] B. Hellouin de Menibus and M. Sablik. Characterisation of sets of limit measures after iteration of a cellular automaton on an initial measure. *CoRR*, abs/1301.1998, 2013.

[FK77] H. Fredricksen and I. J. Kessler. Lexicographic compositions and de-bruijn sequences. *J. Comb. Theory, Ser. A*, 22(1):17–30, 1977.

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

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

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

[Hur90b] M. 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] P. 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] A. Maass. On the sofic limit sets of cellular automata. *Ergodic Theory and Dynamical Systems*, 15:663–684, 7 1995.

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

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