Rice's Theorem for μ-Limit Sets of Cellular Automata

Martin Delacourt*

Laboratoire d'Informatique Fondamentale de Marseille
Université de Provence, France

Abstract. Cellular automata are a parallel and synchronous computing model, made of infinitely many finite automata updating according to the same local rule. Rice's theorem states that any nontrivial property over computable functions is undecidable. It has been adapted by Kari to limit sets of cellular automata [Kar94], that is the set of configurations that can be reached arbitrarily late. This paper proves a new Rice theorem for μ-limit sets, which are sets of configurations often reached arbitrarily late.

1 Introduction

In the field of decidability, a major result is the Rice theorem [Ric53] which states that every property over computable functions is either trivial or undecidable. This being stated for computable functions, it is quite natural to expect a similar result for other computational systems.

In this paper, we will focus on cellular automata, a massively parallel model of computation introduced by von Neumann [vN66]. Cellular automata are composed of infinitely many cells that evolve synchronously following the same local rule. The dynamics of these objects have been well studied, and in particular the notion of limit set i.e. the set of configurations that can be seen arbitrarily late. An important step was achieved by Kari with the equivalent of Rice's theorem for limit sets [Kar94].

Another point of view is to look at configurations that can appear arbitrarily late and often. This supposes to choose the initial configuration according to a measure μ. This approach led to μ-attractors [Hur90] and then to μ-limit sets introduced in [KM00]. Configurations of the μ-limit set are those containing only patterns whose probability does not tend to 0, or equivalently configurations obtained starting from a random initial configuration.

Some results on μ-limit sets are already known, such as the undecidability of the μ-nilpotency [BPT06]. We deal here with Rice's theorem for μ-limit sets. With a construction similar to the one presented in [BDS10] to obtain complex subshifts as μ-limit sets, we will reduce any nontrivial property to the question of μ-nilpotency. First we give some definitions, then two sections are devoted

* Thanks to the project ANR EMC: ANR-09-BLAN-0164.

to the construction of an appropriate cellular automaton, and finally we will be able to prove the reduction.

Due to a lack of space, the proofs are in the appendix.

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 $u = u_0u_1\ldots u_{n-1}$ 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. For $u \in Q^*$ and $0 \le i \le j < |u|$ we define the *subword* $u_{[i,j]} = u_iu_{i+1}\ldots u_j$; this definition can be extended to a configuration $c \in Q^{\mathbb{Z}}$ as $c_{[i,j]} = c_ic_{i+1}\ldots c_j$ for $i, j \in \mathbb{Z}$ with $i \le j$. The *language* of $S \subset Q^{\mathbb{Z}}$ is defined by

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

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

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

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

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

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

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

Definition 1 (Generic configuration). *A configuration is said to be* generic *for an alphabet Q if all words over Q have a strictly positive density in the configuration. If, moreover, any word of length k has density $\frac{1}{|Q|^k}$, the configuration is said to be* normal.

In this paper, we will use a particular normal configuration, that we define here.

Definition 2 (de Bruijn sequence). *A* de Bruijn sequence *of order $n \in \mathbb{N}$ over an alphabet Q is a word of length $|Q|^n + n - 1$ that contains every word of length n as a subword.*

We will consider here a specific de Bruijn sequence of order n noted $DB(n)$ produced as in [FK75].

Definition 3 (de Bruijn configuration). *The* de Bruijn configuration c_{DB} *is the concatenation of de Bruijn sequences of orders n for every $n \in \mathbb{N}$:*

$$c_{DB} = DB(1)DB(2)DB(3)\dots DB(n)\dots$$

Remark 1. Let $n \in \mathbb{N}$ and $u \in Q^n$, we have: $\forall m \in \mathbb{N}, |DB(n+m)|_u = |Q|^m$.

We can prove the following lemma on the regularity of density in prefixes of the de Bruijn configuration.

Lemma 1. *There exists $l_0 \in \mathbb{N}$ such that for all $l \geq l_0$, for any $k \geq |Q|^{2l}$ and any $u \in Q^l$, $\frac{1}{2}d_{c_{DB}}(u) \leq d_{c_{DB[0,k-1]}}(u) \leq 2d_{c_{DB}}(u)$.*

2.2 Cellular Automata

Definition 4 (Cellular automaton). *A* cellular automaton (CA) $\mathcal{A}$ *is a triple* $(Q_\mathcal{A}, r_\mathcal{A}, \delta_\mathcal{A})$ *where $Q_\mathcal{A}$ is a finite set of states called the* alphabet*, $r_\mathcal{A}$ is the* radius *of the automaton, and $\delta_\mathcal{A} : Q_\mathcal{A}^{2r_\mathcal{A}+1} \mapsto Q_\mathcal{A}$ is the* local rule*.*

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

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

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

2.3 μ-Limit Sets

Definition 5 (Uniform Bernoulli measure). *For an alphabet Q, the* uniform Bernoulli measure μ *on configurations over Q is defined by:*

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

.

μ will be the only considered measure through this paper, even though these definitions can be generalized for a large set of measure.

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

Definition 6 (Persistent set). *For a CA $\mathcal{A}$, we define the* persistent set $L_\mu(\mathcal{A}) \subseteq Q^*$ *by: $\forall u \in Q^*$:*

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

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

Remark 2. As said in [KM00], μ-limit sets are closed and shift-invariant. Two μ-limit sets are therefore equal if and only if their languages are equal.

Definition 7 (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. We will reduce all other properties to this problem.

Definition 8 (Set of predecessors). *We define the set of predecessors at time n of a finite word u for a CA $\mathcal{A}$ as $P^n_{\mathcal{A}}(u) = \left\{v \in Q^{|u|+2rn} : \mathcal{A}^n([v]_{-rn}) \subseteq [u]_0\right\}$.*

Remark 3. As we consider the uniform measure μ, $\frac{|P^n_{\mathcal{A}}(u)|}{|Q|^{|u|+2rn}} \to 0 \Leftrightarrow u \notin L_\mu(\mathcal{A})$.

Remark 4. The set of normal configurations has measure 1 in $Q^{\mathbb{Z}}$. Which means that a configuration that is randomly generated according to measure μ is a normal configuration.

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

Lemma 2. *Given a CA $\mathcal{A}$ and a finite word u, for any normal configuration c:*

$$u \in \Lambda_\mu(\mathcal{A}) \Leftrightarrow d_{\mathcal{A}^n(c)}(u) \nrightarrow 0 \text{ when } n \to +\infty$$

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

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

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

2.4 Properties of μ-Limit Sets

Through all this paper, the alphabets of CA will be finite subsets of a countably infinite set $\{\alpha_0, \alpha_1, \alpha_2 \dots\}$. A *property* of μ-limit sets is a family $\mathcal{P}$ of μ-limit sets of CA and any μ-limit set in $\mathcal{P}$ is said to have this property.
A property $\mathcal{P}$ is said to be *nontrivial* if there exist CA $\mathcal{A}_0$ and $\mathcal{A}_1$ such that $\Lambda_\mu(\mathcal{A}_0) \in \mathcal{P}$ and $\Lambda_\mu(\mathcal{A}_1) \notin \mathcal{P}$. For example, μ-nilpotency or the appearance of some state in the μ-limit set are nontrivial properties.

3 Counters

In this section and the following one we describe an automaton $\mathcal{A}_S$, which on normal configurations produces finite segments, separated by a special state #, whose size increases with time. In these segments, we will make computations described in Sect. 5.

We want to erase nearly all of the initial random configuration and only remember some information. This information will be on states $*$ that can appear only at time 0. Each $*$ state sends two counters to its left and right. And these counters will erase everything except a younger counter. Therefore, when two counters meet, they compare their age, and the younger erases the older. If they have the same age, they stop and write a #. The notion of counters was introduced in [DPST10] and used in [BDS10].

Counters are composed of two signals, one faster than the other, and the age of the counter is computed between them. The whole construction (both signals and the binary counter in it) is called a *counter*. Therefore, everything on the initial configuration is erased and forgotten, except for $*$ states. Indeed, even if some counters exist on the initial configuration, they will be older than counters from $*$ when they meet, and hence erased.

For what follows, we will need to have random bits on the #, so we use two states $*_0$ and $*_1$ instead of the unique state $*$. And the bit i is transfered from an $*_i$ to the # produced on its left. We then have $\#_0$ and $\#_1$ instead of #. When it makes no difference, we will speak of # and $*$ for $\#_i$ and $*_i$.

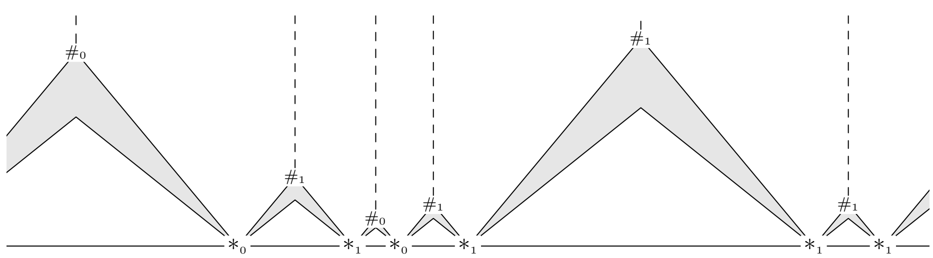

Fig. 1. When two counters launched by a $*$ meet, a # delimiter is produced and counters disappear. Information between $*$ states on the initial configuration is lost when it meets a gray area..

The initialization of a configuration is illustrated in Fig. 1. Counters are schematized by gray areas. # delimiters remain, and we will see later what happens to them.

4 Merging Segments

We saw in Sect. 3, how a special state $*$ on the initial configuration gave birth to counters protecting everything inside them until they meet some other counter born the same way. In this section, we will describe the evolution of the automaton $\mathcal{A}_S$ after this time of initialization. When two counters of the same age meet, they disappear and a # is produced.

Definition 9 (Segment). *A* segment *u is a subword of a configuration delimited by two # and containing no # inside* $\left(u \in \#\,(Q \setminus \{\#\})^*\,\#\right)$. *The* size *of a segment is the number of cells between both #.*

When a # is produced in automaton $\mathcal{A}_S$, it sends a signal on its right to detect the first # on its right. If the signal catches the inside of a counter still in activity before reaching a #, it waits until this counter produces a #. Then both # have recognised each other and the segment between them becomes "conscious". It launches a computation inside itself, and this will be the concern of Sect. 5. But as we will need arbitrarily large space for computation, we will remove small segments and replace them by larger ones. Therefore, at some times, we will erase some #, and the segments that were formerly separated will join their space. We describe here mechanisms that lead to this merging process, and then see how it behaves with μ-limit sets. This happens inside any segment, and in parallel with the computation from Sect. 5.

A segment is said to be *well-formed* if it is delimited by two # that have themselves been created by $*$ states on the initial configuration. To construct $\mathcal{A}_S$, we additionally attribute a color to each segment. There will be Red (R) and Blue (B) segments. So we will have 4 states to replace $*_0$ and $*_1$: $*_{r0}$, $*_{r1}$, $*_{b0}$ and $*_{b1}$. We still use $*$ to refer to any of them indistinctly. An initial segment has color R when produced by $*_{r0}$ or $*_{r1}$, and else color B. The random bit is still transfered to the # on the left as described in Sect. 3.

We require that any segment stores and updates its age since the initial configuration. We'll add two counters (one on each side of the segment) to perform this task. Moreover, we want to ensure that the storage of the age of a well-formed segment does not need more than $\lfloor\sqrt{(n)}\rfloor$ cells where n is the size of the segment. This means we need to know the size of the segment. This can be computed and stored with space $\log(n)$. To maintain the property, segments will merge when the age becomes too large for them.

Suppose we use an alphabet of size $K \geq 3$ to store the age, then the space used in a segment becomes too large at some time K^i with $i \in \mathbb{N}$ (when $i+1 \geq \lfloor\sqrt{(n)}\rfloor$). Every segment has to decide whether it will need to merge, and to tell its neighbors before time K^i. At that time, any segment that needs more space will merge according to the following conditions:

1. if none of its neighbors want to merge, it merges with the left one,
2. if one only of its neighbors wants to merge, it merges with that one,

3. if both its neighbors want to merge, it merges with the left one except if this neighbor has the same color, and the right one has the other color.

Remark 5. Each segment can decide in $(i+1)^2$ steps, if it is larger than $(i+1)^2$. Then it can write on each side if it wants to merge at time K^i or not in less than $2(i+1)^2$ steps. If a segment wants to merge, it can check its neighbors' will on both sides and decide its own behavior in $(i+1)^2$.

The # between two segments that merge together is erased with the age counters around it. Then another cycle starts on the left side of the new segment. Many successive # have possibly disappeared, so the merging is not necessarily two segments becoming one but many segments becoming one. As at least one # has been erased inside the new segment, we use the bit from the leftmost $\#_i$ erased to determine the color of the new segment. If $i = 0$, it will be R, and else B.

We call *initial* segment, a well-formed segment such that only one cell inside it contained a $*$ on the initial configuration. That is, a segment that is well-formed and not created from a merging. And we call *successor* segment, a segment well-formed but not initial, that is, created by a merging of well-formed segments that are its *predecessors*. We so define a set of predecessors at each time.

Remark 6. When two segments merge at time K^i, at least one of them wanted to merge, which means one of them was smaller than $(i+1)^2$.

If three or more segments merge at time K^i, they all wanted to merge, so they were all smaller than $(i+1)^2$.

$\forall i \in \mathbb{N}$, at time $t > K^i$, any segment has a size greater than $(i+1)^2$. This is clear since any segment smaller would have merged at time K^i.

Remark 7. Red and blue segments are initially randomly distributed according to the uniform measure. When some segments merge, the new color is chosen independently from the colors of predecessors or neighbors, and only according to a random bit, so the distribution of colors remains random.

The general behavior of the segments among themselves is illustrated in Fig. 2.

Thanks to the following claim, we will be able to prove the first important result for automaton $\mathcal{A}_S$: Prop. 1, which says that # states tend to be sparse enough to be left outside of $\Lambda_\mu(\mathcal{A}_S)$.

Claim. The density of cells outside well-formed segments on a normal configuration tends to 0 as time passes.

Proposition 1. *There is no # in the μ-limit set of $\mathcal{A}_S$.*

This comes from the fact that well-formed segments tend to cover the image of a normal configuration. As their growth is permanent, # states are eventually separated by arbitrarily large words.

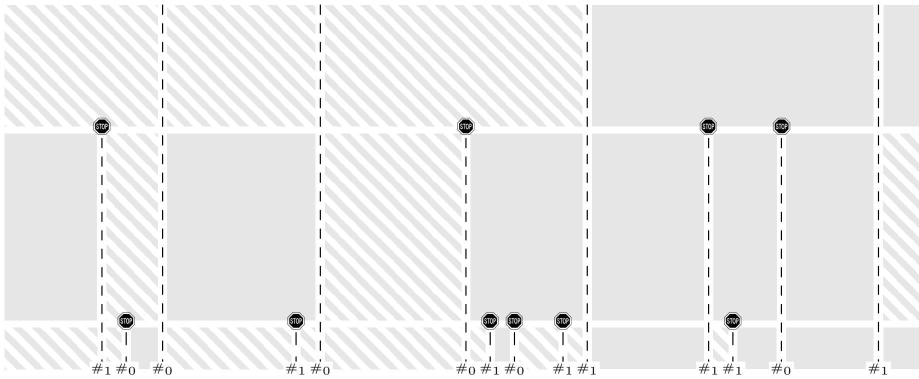

Fig. 2. A # remains until two segments merge. Blue segments are in plain gray, red ones in hashed gray. At time K^i, small segments merge together or with their left neighbor.

In Sect 5, we will need to have an upper bound on the size of a large proportion of segments. We will prove such a bound in the following lemma.

A segment is said to be *acceptable* if it is well-formed and if its size is $n \leq K^{i/4}$ at time $K^i \leq t < K^{i+1}$. In the sequel, we consider the automaton on a normal configuration c_N.

We will show that large segments exist with low probability. First we use the merging protocol, and the colors to justify that a lot of segments rarely merge all together. Then, we show that large initial segments are quite unlikely. In both cases, we give bounds on the probability of large segments.

We can now prove the next lemma:

Lemma 3. *The density of non acceptable segments tends to* 0 *as time passes.*

To do this, each non acceptable segment is seen as either initial, the product of a merging, or a successor of some non acceptable segment. In the third case, we consider the first predecessor that was in another case and show that it concerns few segments only.

Denote $S_t, t \in \mathbb{N}$ the set of acceptable segments successors of acceptable segments at time t. This set contains all possible segments, it is finite for any time, and it does not depend of the initial configuration.

Remark 8. The same proof as for previous lemma shows that $d_{c_N}(S_t) \to_{t\to\infty} 1$.

Thanks to this, we only need to look at the behavior of the automaton inside segments of S_t, the words that will remain in the μ-limit set will be the words that appear often in these segments.

5 Rice Theorem

The idea here is to copy the principle of the proof of Rice's theorem for limit sets from [Kar94]. We want to reduce any nontrivial property over μ-limit sets to the μ-nilpotency of a CA $\mathcal{H}$ of radius 1 having a permanent state q.

First we construct an automaton to prove the following proposition:

Proposition 2. *For any CA $\mathcal{H}$ of radius 1, where a state q is permanent, and CA $\mathcal{A}$, there exists a CA $\mathcal{B}$ such that:*

- *if $\mathcal{H}$ is μ-nilpotent, $\Lambda_\mu(\mathcal{B}) = \Lambda_\mu(\mathcal{A})$.*
- *if $\mathcal{H}$ is not μ-nilpotent, $\Lambda_\mu(\mathcal{B}) = \alpha^{\mathbb{Z}}$, for some chosen $\alpha \in Q_{\mathcal{A}}$.*

Then, given a property $\mathcal{P}$, for two automata $\mathcal{A}_1$ and $\mathcal{A}_2$, such that one exactly of $\Lambda_\mu(\mathcal{A}_1)$ and $\Lambda_\mu(\mathcal{A}_2)$ has property $\mathcal{P}$, we construct $\mathcal{B}_1$ and $\mathcal{B}_2$. If there existed an algorithm to decide whether a μ-limit set has property $\mathcal{P}$, we could use this algorithm with $\mathcal{B}_1$ and $\mathcal{B}_2$. If only one among $\Lambda_\mu(\mathcal{B}_1)$ and $\Lambda_\mu(\mathcal{B}_2)$ has property $\mathcal{P}$, they have different μ-limit sets and $\mathcal{H}$ is μ-nilpotent. In the other case, their μ-limit sets cannot be $\Lambda_\mu(\mathcal{A}_1)$ and $\Lambda_\mu(\mathcal{A}_2)$, and $\mathcal{H}$ is not μ-nilpotent.

5.1 Construction

The automaton $\mathcal{B}$ is based upon $\mathcal{A}_S$ from previous sections. The whole construction of counters and segments is exactly as described in Sect(s). 3 and 4. We now describe the computation in a segment. We will only concern ourselves with well-formed segments as, for a normal configuration, every cell is eventually reached by such a segment. We partially test the μ-nilpotency of $\mathcal{H}$ in every segment, if the test is conlusive, we simulate $\mathcal{A}$, meaning we write an image of a prefix of the de Bruijn configuration, if not, we write the uniform word α^n.

Remark 9. We will use the de Bruijn configuration c_{DB} over the alphabet $Q_{\mathcal{A}}$ defined in 2.1. Thanks to [FK75], a de Bruijn sequence of order k can be computed in space $O(k)$ and time $O(|Q|^k)$. Therefore we can fill a segment of length n with a prefix of c_{DB} in space $O(\log(n))$ and time $O(n)$.

Remark 10. As q is a permanent state in the radius 1 CA $\mathcal{H}$, it behaves like a *wall*, which means no information can travel through a q state. So a simulation of $\mathcal{H}$ over a word $u \in q\left(Q_{\mathcal{H}} \setminus \{q\}\right)^l q$ needs only space l for any l.

In any well-formed segment of size $n \in \mathbb{N}$, the computation of a Turing machine starts on the left of the segment at every time K^i for $i \in \mathbb{N}$.

1. it measures and stores on each side the segment's size.
2. it simulates $\mathcal{H}$ on every $u \in q\left(Q_{\mathcal{H}} \setminus \{q\}\right)^l q$ for $l \leq \frac{1}{2}\left(\log_{|Q_{\mathcal{H}}|}(K^{i/4})\right) - 1$ during $|Q_{\mathcal{H}}|^l$ timesteps. This is how we test the μ-nilpotency of $\mathcal{H}$. If one of the computed images is not q^l, the segment does not simulate $\mathcal{A}$. If all the images are q^l, the segment simulates $\mathcal{A}$.
3. on the left of the segment, it computes $j(i) = \lfloor \log(\log(i)) \rfloor$.

4. if the segment simulates $\mathcal{A}$, the machine computes a prefix of length n of c_{DB}, then computes and writes its $j(i)$-th image by $\mathcal{A}$. If the segment does not simulate $\mathcal{A}$, the head writes α^n over the segment.
5. the machine stops when the whole computation and writing is over or when it has reached time K^{i+1}. At that time, the machine erases itself, leaving what was written.

Remark 11. Each cell in a segment contains a couple:

- a state for computation, storage of the age or the length of the segment,
- a state from $Q_{\mathcal{A}}$.

The first state (computation) is for most cells left blank, then the couple is seen as a state of $Q_{\mathcal{A}}$.

Remark 12. The machine needs only $O(j(i) + \log(n))$ cells to compute. There are $O(\sqrt{(}n))$ additional cells used to count the age on each side, and $O(1)$ cells for signals moving through the segment. And only $O(j(i)n)$ steps are required to perform it.

When a segment is formed by a merging, the data of its cells (on the first layer) is not removed until a new state of $Q_{\mathcal{A}}$ has to be written.

To prove Prop. 2, we will need two lemmas:

Lemma 4. *There exists i_0 such that, for $i \geq i_0$, the computation is finished before $K^{i+1} - 1$ in every segment that is acceptable at time K^i.*

This is easily proved, since the length of acceptable segments is bounded.

And the second lemma, that will let us test the μ-nilpotency of $\mathcal{H}$:

Lemma 5. *If $\mathcal{H}$ is not μ-nilpotent, there exists $l \in \mathbb{N}$ and $u \in q(Q_{\mathcal{H}} \setminus \{q\})^l q$ such that $\mathcal{H}^{|Q_{\mathcal{H}}|^l}(u) \neq q^{l+2}$.*

To prove it, we just use the fact that words which cannot have any permanent state in their antecedents cannot be persistent, since the measure of their predecessors' set tends to 0.

5.2 $\mathcal{H}$ μ-Nilpotent

In this section, we suppose $\mathcal{H}$ is μ-nilpotent and we will show $\Lambda_\mu(\mathcal{B}) \subseteq \Lambda_\mu(\mathcal{A})$.

First, we make sure that the simulation happens everywhere in this case.

Claim. If $\mathcal{H}$ is μ-nilpotent, every well-formed segment simulates $\mathcal{A}$.

Then we prove the following lemma:

Lemma 6. *If $\mathcal{H}$ is μ-nilpotent, then $\Lambda_\mu(\mathcal{B}) \subseteq \Lambda_\mu(\mathcal{A})$*

The proof needs a description of the content of a segment with computation parts, and words computed by the simulation of $\mathcal{A}$.

Remark 13. Thanks to Lemma 4, any $s \in S_t$ with $|s| = n$ at time $K^i \leq t < K^{i+1}$ contains:

- $O(\sqrt(n))$ cells for computation, they will not appear in $L_\mu(\mathcal{B})$.
- a subword of $\mathcal{A}^{j(i)}(c_{DB[0..n-1]})$ computed between times K^i and t.
- a concatenation of subwords of $\mathcal{A}^{j(i-1)}(c_{DB[0..n-1]})$ computed before time K^i that were not erased during the merging.

To prove the lemma, we show that any word of $L_\mu(\mathcal{B})$ appears often in large acceptable segments. Then we prove that if a word appears often in an acceptable segment, thanks to the previous remark, it appears often in at least an image of c_{DB} by $\mathcal{A}$. As c_{DB} is normal, we can conclude.

Now we show the second inclusion:

Lemma 7. *If $\mathcal{H}$ is μ-nilpotent, then $\Lambda_\mu(\mathcal{A}) \subseteq \Lambda_\mu(\mathcal{B})$*

To prove this lemma, we consider a word in $L_\mu(\mathcal{A})$. There exists necessarily a sequence of images $(\mathcal{A}^{t_x}(c_{DB}))_{x\in\mathbb{N}}$ in which the density of u does not tend to 0. We take a sequence $(\tau_x)_{x\in\mathbb{N}}$ of times at which $\mathcal{B}^{\tau_x}$ simulates $\mathcal{A}^{t_x}$. Then, thanks to the regularity property of c_{DB}, we can prove that the density of u in the images of a normal configuration by $\mathcal{B}$ does not tend to 0.

5.3 $\mathcal{H}$ not μ-Nilpotent

In this case, we first make sure that after some time, $\mathcal{A}$ is not simulated in any segment anymore.

Claim. If $\mathcal{H}$ is not μ-nilpotent, there exists i_0 such that, for all $i \geq i_0$, no well-formed segment at time $K^i \leq t < K^{i+1}$ simulates $\mathcal{A}$.

Then we conclude by saying that any letter different from α has a density that tends to 0. Which forces $\alpha^\mathbb{Z}$ to be the unique configuration in $L_\mu(\mathcal{B})$.

Lemma 8. *If $\mathcal{H}$ is not μ-nilpotent, then $\alpha^* = L_\mu(\mathcal{B})$.*

This ends the proof of Prop. 2.

5.4 Rice Theorem

First we need to consider two automata $\mathcal{A}_1$ and $\mathcal{A}_2$ over the same alphabet:

Lemma 9. *For any nontrivial property, there exist CA $\mathcal{A}_0$ and $\mathcal{A}_1$ over the same alphabet $Q_\mathcal{A}$ such that one among $\Lambda_\mu(\mathcal{A}_0)$ and $\Lambda_\mu(\mathcal{A}_1)$ has this property, and the other not.*

We can prove this lemma by taking multiple copies of each state of both automata, in order to get an alphabet which size is the least common multiple of both sizes.

And finally, we complete the proof of the theorem:

Theorem 1. *Any nontrivial property of μ-limit sets of cellular automata is undecidable.*

As announced, from $\mathcal{A}_1$ and $\mathcal{A}_2$, we construct $\mathcal{B}_1$ and $\mathcal{B}_2$, then deciding a property $\mathcal{P}$ leads to deciding the μ-nilpotency of $\mathcal{H}$.

6 Conclusion

The result presented in this article is that no algorithmic property over μ-limit sets can be decided, except for trivial ones. This, as [BDS10], shows the complexity and hence the interest of this object. We have the same restriction as in [Kar94], that is, we work on an unlimited set of states. One property at least becomes decidable if we limit the set of possible states, it is having the fullshift as μ-limit set. Which is equivalent to being surjective.

In [GR10], it was proved that surjectivity was the only decidable problem on limit sets with a fixed alphabet. This extension could perhaps be adapted to μ-limit sets and then show another parallel between limit and μ-limit sets.

This is also another use of counters and segments, showing how powerful this tool can be for cellular automata. Especially concerning μ-limit sets.

References

[BDS10] Laurent Boyer, Martin Delacourt, and Mathieu Sablik. Construction of μ-limit sets. *Journées Automates Cellulaires*, 76–87, 2010.

[BPT06] Laurent Boyer, Victor Poupet, and Guillaume Theyssier. On the Complexity of Limit Sets of Cellular Automata Associated with Probability Measures. *MFCS 2006*, LNCS 4162:190–201, 2006.

[DPST10] Martin Delacourt, Victor Poupet, Mathieu Sablik, and Guillaume Theyssier. Directional Dynamics along Arbitrary Curves in Cellular Automata. *Theoretical Computer Science*, A paraître, 2010.

[FK75] Harold Fredricksen and Irving Kessler. Lexicographic compositions and de-Bruijn sequences. *Journal of combinatorial theory*, A 22:17-30, 1977

[GR10] Pierre Guillon and Gaétan Richard. Revisiting the Rice Theorem for Cellular Automata. *STACS, 2010*, pages 441–452, 2010.

[Hur90] Mike Hurley. Ergodic aspects of cellular automata. *Ergodic Theory Dynam. Systems*, 10(4):671–685, 1990.

[Kar94] Jarkko Kari. Rice's Theorem for the Limit Sets of Cellular Automata. *Theor. Comput. Sci.*, 127(2):229–254, 1994.

[KM00] Petr Kurka and Alejandro Maass. Limit sets of cellular automata associated to probability measures. *Journal of Statistical Physics*, 100(5):1031–1047, 2000.

[Ric53] Henry Gordon Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*, 74(2):358–366, 1953.

[vN66] John von Neumann. Theory of Self-Reproducing Automata. University of Illinois Press, Champaign, IL, USA, 1966.

A Proofs

A.1 Proofs for Sect. 2

Lemma 1. *There exists* $l_0 \in \mathbb{N}$ *such that for all* $l \geq l_0$*, for any* $k \geq |Q|^{2l}$ *and any* $u \in Q^l$*,* $\frac{1}{2} d_{c_{DB}}(u) \leq d_{c_{DB[0,k-1]}}(u) \leq 2 d_{c_{DB}}(u)$.

Proof. Let $l \in \mathbb{N}$. The length of the concatenation of the l first de Bruijn sequences is: $s(l) = \sum_{i=1}^{l}(|Q|^i + i - 1) = \frac{|Q|^{l+1}-1}{|Q|-1} + \frac{l(l-1)}{2}$. Let $k \geq s(l)$, there exists $m \in \mathbb{N}$ such that $s(l+m) \leq k \leq s(l+m+1)$. Now $d_{c_{DB[0,k-1]}}(u) \geq \frac{1}{k}\left(\sum_{i=l}^{l+m} |DB(i)|_u\right)$ and there are at least $|Q|^i$ words of size $l+i$ containing u. Therefore $d_{c_{DB[0,k-1]}}(u) \geq \frac{\sum_{i=0}^{m} |Q|^i}{s(l+m+1)}$. And $\frac{\sum_{i=0}^{m} |Q|^i}{s(l+m+1)} \to_{m\to\infty} \frac{1}{|Q|^l}$.

As $d_{c_{DB}}(u) = \limsup_{k\to+\infty} \frac{|c_{DB[0,k-1]}|_u}{k-|u|}$, we have proved that $d_{c_{DB}}(u) \geq \frac{1}{|Q|^l}$, but as $\sum_{u \in Q^l} d_{c_{DB}}(u) = 1$, this means that c_{DB} is effectively a normal configuration.

Then, there exists l_0 such that $\forall l \geq l_0, k \geq |Q|^{2l}$, $\frac{1}{2|Q|^l} \leq d_{c_{DB[0,k-1]}}(u) \leq \frac{2}{|Q|^l}$. □

Lemma 2. *Given a CA* $\mathcal{A}$ *and a finite word* u*, for any normal configuration* c*:*

$$u \in \Lambda_\mu(\mathcal{A}). \Leftrightarrow d_{\mathcal{A}^n(c)}(u) \nrightarrow 0 \text{ when } n \to +\infty.$$

Proof. Actually, we prove here that for any $n \in \mathbb{N}$:
$d_{\mathcal{A}^n(c)}(u) = \mathcal{A}^n\mu(u) = \frac{|P^n_{\mathcal{A}}(u)|}{|Q|^{|u|+2rn}}$.
The second equality is clear.
Let $n \in \mathbb{N}$. We note r the radius of $\mathcal{A}$. Since any occurence of u in $\mathcal{A}^n(c)$ corresponds to an occurence of a predecessor of u in c :

$$d_{\mathcal{A}^n(c)}(u) = \lim_{k\to+\infty} \frac{|\mathcal{A}^n(c)_{[-k,k]}|_u}{2k+1-|u|} = \lim_{k\to+\infty} \sum_{v \in P^n_{\mathcal{A}}(u)} \frac{|c_{[-k-rn,k+rn]}|_v}{2k+2rn+1-(|u|+2rn)}.$$

And as c is normal, for any $v \in P^n_{\mathcal{A}}(u) : |c_{[-k-rn,k+rn]}|_v \sim_{k\to+\infty} \frac{2k+1}{|Q|^{|u|+2rn}}$.
Then:

$$\begin{aligned} d_{\mathcal{A}^n(c)}(u) &= \sum_{v \in P^n_{\mathcal{A}}(u)} \lim_{k\to+\infty} \left(\frac{1}{2k+1-|u|} \frac{2k+1}{|Q|^{|u|+2rn}} \right) \\ &= \sum_{v \in P^n_{\mathcal{A}}(u)} \frac{1}{|Q|^{|u|+2rn}} \\ &= \frac{|P^n_{\mathcal{A}}(u)|}{|Q|^{|u|+2rn}}. \end{aligned}$$

□

A.2 Proofs for Sect. 4

Claim. The density of cells outside well-formed segments on a normal configuration tends to 0.

Proof. The proof is clear since such a cell needs predecessors without states $*$ on each side in the initial configuration. □

Proposition 1. *There is no # in the μ-limit set of $\mathcal{A}_S$.*

Proof. Let $\epsilon > 0$. We consider a normal configuration. previous claim tells that there exists $t_0 \in \mathbb{N}$ such that $\forall t > t_0$, the density of cells outside well-formed segments is less than $\epsilon/2$. Take i_0 such that $\forall i > i_0, 1/(i+1)^2 \leq \epsilon/2$, thanks to remark 6, for $t > t_1$ with $t_1 = max(t_0, K^i)$, the size of any segment is bigger than $(i+1)^2$. For $t > t_1$, as the # have to come either from well-formed segments (density less than $1/(i+1)^2$), or from the rest of the configuration (density less than $\epsilon/2$), the total density of # is less than ϵ.

And finally, this density tends to 0 and Lemma 2 concludes. □

Claim. A segment produced by the merging of three or more segments at time K^i has size n with probability less than $\frac{4n}{(i+1)^2}2^{-\frac{n}{(i+1)^2}}$.

Proof. Consider such a segment of size n. All the predecessors of it were smaller than $(i+1)^2$, or they would not have merged together. So $\frac{n}{(i+1)^2}$ segments at least have merged. Now consider the colors of these segments. As they prefer to merge with a segment of the same color, and if not possible on their left, the colors' distribution among them was: $R(RB)^lR^k$, $(RB)^lR^k$, $R(RB)^lB^k$ or $R(RB)^lB^k$ (or symmetrically if starting with B) for some $k, l \in \mathbb{N}$. So the distribution is determined by its shape (among four possible shapes), its length and k. Therefore the probability of such a succession of segments is less than $4\frac{n}{(i+1)^2}2^{-\frac{n}{(i+1)^2}}$.

□

Claim. An initial segment has length n with probability less than $n\left(\frac{2}{q}\right)^3\left(\frac{q-2}{q}\right)^{2n}$ where q is the number of states of the automaton.

Proof. An initial segment of length n is produced by three $*$ distant from l_1 and l_2, with $(l_1 + l_2)/2 = n$. So the probability is less than $\frac{2}{q}\left(\frac{q-2}{q}\right)^{l_1}\frac{2}{q}\left(\frac{q-2}{q}\right)^{l_2}\frac{2}{q}$. Considering the n possibilities for the choice of l_1 and l_2, we have $n\left(\frac{2}{q}\right)^3\left(\frac{q-2}{q}\right)^{2n}$

□

Lemma 3. *The density of non acceptable segments tends to* 0 *as time passes.*

Proof. Consider time $K^i \leq t < K^{i+1}$. Thanks to remark 6, and for large enough i (such that $K^{i/4} \geq 2i^2$), if a segment is larger than $K^{i/4}$ at time t, then it is:

1. either an initial segment.
2. either a segment produced by the merging of many segments.

3. or the successor of a segment that was larger than $K^{(i-1)/4}$ at time K^{i-1}, and that possibly merged with a segment smaller than i^2 at time K^i.

For case 3, we consider the chain of non acceptable predecessors of the segment at time K^j, $j \leq i$. The oldest predecessor of this chain is either case 1 or case 2: there exists $h \leq i$ minimal such that at time K^j, for all $h \leq j \leq i$, one predecessor of the segment is larger than $K^{j/4}$. If this predecessor's size at K^h was n, its size at K^i is less than $n + \sum_0^i j^2$ since it merged with at most one small segment at each K^j.

There exists i_0 such that $\forall i \geq i_0$, $\sum_0^i j^2 \leq K^{i/4-1}$, therefore, as $n + \sum_0^i j^2 \geq K^{i/4}$, we have $n \geq K^{i/4-1}$. The predecessor was either case 1 or case 2 and its size doubled at most between steps h and i. We treat both cases (with $h \leq i$) thanks to the two previous claims.

So the density d_i of cells in segments larger than $K^{i/4}$ at time K^i is the sum of:

- cells in initial segments larger than $K^{i/4-1}$, that at most doubled;
- cells in case 2 segments created before i and that at most doubled.

$$d_i \leq \sum_{n \geq K^{i/4-1}} 2n \left(n \left(\frac{2}{q}\right)^3 \left(\frac{q-2}{q}\right)^{2n} \right) + \sum_{h \leq i} \sum_{n \geq K^{i/4-1}} 2n \left(\frac{4n}{(h+1)^2} 2^{-\frac{n}{(h+1)^2}} \right)$$

.

$$d_i \leq \sum_{n \geq K^{i/4-1}} 2n \left(n \left(\frac{2}{q}\right)^3 \left(\frac{q-2}{q}\right)^{2n} \right) + i \sum_{n \geq K^{i/4-1}} 2n \left(4n \times 2^{-\frac{n}{(i+1)^2}} \right)$$

Finally, $d_i \to 0$. □

A.3 Proofs for Sect. 5

Lemma 4. *There exists i_0 such that, for $i \geq i_0$, the computation is finished before $K^{i+1} - 1$ in every segment that is acceptable at time K^i.*

Proof. As said in remark 12, the computation needs $O(j(i)n) = O(n \log(\log(i)))$, and as $n \log(\log(i)) = o(K^i)$ for $n \leq K^{i/4}$, there exists i_0 such that the computation is finished within K^i steps for acceptable segments. And then, at time $K^{i+1} - 1 \geq K^i + K^i$, the computation is over. □

Lemma 5. *If $\mathcal{H}$ is not μ-nilpotent, there exists $l \in \mathbb{N}$ and $u \in q(Q_{\mathcal{H}} \setminus \{q\})^l q$ such that $\mathcal{H}^{|Q_{\mathcal{H}}|^l}(u) \neq q^{l+2}$.*

Proof. As q is a wall for $\mathcal{H}$, the behavior between two q is ultimately periodic. If any word in these periods is uniform, non uniform words will disappear as time passes.

Suppose that for any $u \in q(Q_{\mathcal{H}} \setminus \{q\})^* q$, $\mathcal{H}(u) = q^{|u|+2}$. Now consider $v \in L_\mu(\mathcal{H})$. Suppose for all $x \in [t - \log(\log(t)), t]$ and $y \in [t + |v|, t + |v| + \log(\log(t))]$, for any u of length $|v| + 2t$ with $u_x = q = u_y$, we have $\mathcal{H}^t(u) \neq v$. In this case, the density of predecessors of v is 0.

Therefore, there exists $u \in Q_{\mathcal{H}}^{|v|+2t}$ such that $\mathcal{H}^t(u) = v$ and $u_x = q = u_y$ with $x \in [t - \log(\log(t)), t]$ and $y \in [t + |v|, t + |v| + \log(\log(t))]$.

Then, as $y - x \leq |v| + 2\log(\log(t))$, and for large enough $t \geq |Q_{\mathcal{H}}|^{|v|+2\log(\log(t))}$, $v = \mathcal{H}^t(u) = q^{|v|}$ by hypothesis. And $\mathcal{H}$ is μ-nilpotent. □

Claim. If $\mathcal{H}$ is μ-nilpotent, every well-formed segment simulates $\mathcal{A}$.

Proof. Let s a well-formed segment. It simulates $\mathcal{H}$ on every $u \in q\,(Q_{\mathcal{H}} \setminus \{q\})^p\, q$ during $|Q_{\mathcal{H}}|^p$ for a finite number of values for p. Thanks to Lemma 5, every such simulation computes q^{p+2}, so s decides to simulate $\mathcal{A}$. □

Claim. Let $u \in L_\mu(\mathcal{B})$ ($|u| = k$) and $\epsilon > 0$. Let $s \in S_t$ for $K^i \leq t < K^{i+1} (i \in \mathbb{N})$ such that $|Q_{\mathcal{A}}|^{2(k+2j(i))} \leq (i-1)^2$ and $d_s(u) \geq \epsilon$, then:

- either $d_{\mathcal{A}^{j(i-1)}(c_{DB})}(u) \geq \frac{\epsilon}{4}$,
- or $d_{\mathcal{A}^{j(i)}(c_{DB})}(u) \geq \frac{\epsilon}{4}$

Proof. If the computation is over in the segment, as $|s| \geq |Q_{\mathcal{A}}|^{2(k+2j(i))}$, Lemma 1 applies for antecedents of u by $\mathcal{A}^{j(i)}$ and:

$$d_s(u) \leq d_{c_{DB[0,|s|]}}(P_{\mathcal{A}}^{j(i)}(u)) \leq 2d_{c_{DB}}(P_{\mathcal{A}}^{j(i)}(u)) \leq 2d_{\mathcal{A}^{j(i)}(c_{DB})}(u).$$

If the computation is not even begun, s contains the concatenation of words written by the computation in its predecessors. As they were all larger than $(i-1)^2 \geq |Q_{\mathcal{A}}|^{2(k+2j(i-1))}$, the same reasoning shows that the density of u in each of them was less than $2d_{\mathcal{A}^{j(i-1)}(c_{DB})}(u)$. So $d_s(u) \leq 2d_{\mathcal{A}^{j(i-1)}(c_{DB})}(u)$.

And finally, we conclude since the density is less than the sum of both:

$$d_s(u) \leq 2d_{\mathcal{A}^{j(i)}(c_{DB})}(u) + 2d_{\mathcal{A}^{j(i-1)}(c_{DB})}(u).$$

□

Lemma 6. *If $\mathcal{H}$ is μ-nilpotent, then $\Lambda_\mu(\mathcal{B}) \subseteq \Lambda_\mu(\mathcal{A})$*

Proof. We will prove $L_\mu(\mathcal{B}) \subseteq L_\mu(\mathcal{A})$.

Let $u \in L_\mu(\mathcal{B})$ ($|u| = k$), there exists $\epsilon > 0$ and $(t_x)_x$ such that $\forall x \in \mathbb{N}, d_{\mathcal{B}^{t_x}(c_N)}(u) \geq \epsilon$. Thanks to Lemma 3 and remark 8, $d_{\mathcal{B}^t(c_N)}(S_t) \to 1$, so there exists $x_0 \in \mathbb{N}$ such that $\forall x \geq x_0$, there exists an acceptable segment $s_x \in S_{t_x} \cap L\left(\mathcal{B}^{t_x}(c_N)\right)$ with $d_{s_x}(u) \geq \epsilon$.

There exists $i_0 \in \mathbb{N}$ such that for all $i \geq i_0 - 1$, $|Q_{\mathcal{A}}|^{2(k+2j(i))} \leq i^2$. Consider $x_1 \geq x_0$ minimal such that $t_{x_1} \geq K^{i_0}$. Now previous claim ensures that there exists $j_x \in \{j(i-1), j(i)\}$ such that $d_{\mathcal{A}^{j_x}(c_{DB})}(u) \geq \frac{\epsilon}{4}$. As $j_x \to_{x\to\infty} \infty$, $u \in L_\mu(\mathcal{A})$.

□

Lemma 7. *If $\mathcal{H}$ is μ-nilpotent, then $\Lambda_\mu(\mathcal{A}) \subseteq \Lambda_\mu(\mathcal{B})$*

Proof. Let $u \in L_\mu(\mathcal{A})$ ($|u| = k$). There exists $(t_x)_x$ such that $d_{\mathcal{A}^{t_x}(c_{DB})}(u) \nrightarrow_{x\to\infty} 0$. Denote $i_x = K^{K^{t_x}}$, clearly, there exists $x_0 \in \mathbb{N}$ such that $\forall x \geq x_0$, $i_x^2 \geq |Q_\mathcal{A}|^{2(k+2t_x)}$. Now, denote $\tau_x = K^{i_x+1} - 1$, at time τ_x, any acceptable segment of the automaton $\mathcal{B}$ contains the image by $\mathcal{A}^{t_x}$ of a prefix of c_{DB}.

For any $x \geq x_0$, $d_{\mathcal{B}^{\tau_x}(c_N)}(u) \geq d_{\mathcal{B}^{\tau_x}(c_N)}(S_{\tau_x}) \min_{s\in S_{\tau_x}} d_s(u)$.

Thanks to Lemma 3, $d_{\mathcal{B}^{\tau_x}(c_N)}(S_{\tau_x}) \to 1$.

Now, consider an acceptable segment s at time τ_x. The computation in s is over and the word $\mathcal{A}^{t_x}(c_{DB})[0, |s|-1]$ is written. As remarked in 12, there exists $\beta > 0$ such that less than $\beta_s = \beta\frac{\sqrt(|s|)}{|s|}$ cells are used for computation in s.

We have $d_s(u) \geq d_{\mathcal{A}^{t_x}(c_{DB[0,|s|-1]})}(u) - \beta_s$. And $d_s(u) \geq d_{c_{DB[0,|s|-1]}}(P_\mathcal{A}^{t_x}(u)) - \beta_s$.

As s is acceptable, $|s| \geq i_x^2 \geq |Q_\mathcal{A}|^{2(k+2t_x)}$, so, we can apply Lemma 1 with words in $P_\mathcal{A}^{t_x}(u)$, which are all $k+2t_x$ long: $d_{c_{DB[0,|s|-1]}}(P_\mathcal{A}^{t_x}(u)) \geq \frac{1}{2} d_{c_{DB}}(P_\mathcal{A}^{t_x}(u))$. Therefore $d_s(u) \geq \frac{1}{2} d_{\mathcal{A}^{t_x}(c_{DB})}(u) - \beta_s$.

And thanks to remark 6, $|s| \geq (i_x+1)^2$, hence $\beta_s \leq \frac{\beta}{i_x+1}$.

We finally have:

$$d_{\mathcal{B}^{\tau_x}(c_N)}(u) \geq d_{\mathcal{B}^{\tau_x}(c_N)}(S_{\tau_x}) \left(\frac{1}{2} d_{\mathcal{A}^{t_x}(c_{DB})}(u) - \frac{\beta}{i_x+1}\right).$$

Therefore $d_{\mathcal{B}^{\tau_x}(c_N)}(u) \nrightarrow_{x\to\infty} 0$ and $u \in L_\mu(\mathcal{B})$.

□

Claim. If $\mathcal{H}$ is not μ-nilpotent, there exists i_0 such that, for all $i \geq i_0$, no well-formed segment at time $K^i \leq t < K^{i+1}$ simulates $\mathcal{A}$.

Proof. Thanks to 5, there exists $l \in \mathbb{N}$ and $u \in Q_\mathcal{H}^l$ such that $\mathcal{H}^{|Q_\mathcal{H}|^l}(u) \neq q^l$. There exists i_0 such that $l \leq \frac{1}{2}\left(\log_{|Q_\mathcal{H}|}(K^{i_0/4})\right) - 1$. Then for any well-formed segment at time $K^{i_0} \leq t$, the simulation of u is done and the segment does not simulate $\mathcal{A}$. □

Lemma 8. *If $\mathcal{H}$ is not μ-nilpotent, then $\alpha^* = L_\mu(\mathcal{B})$.*

Proof. For each segment of length n, there are $O(\sqrt(n))$ cells used for computation and signals. As the length of the segments tends to infinity, the density of computation cells tends to 0. So they do not appear in $L_\mu(\mathcal{B})$. Thanks to the previous claim, we know that α is written all over the segment, therefore other cells can only contain α. So any word in $L_\mu(\mathcal{B})$ is in α^*. □

Lemma 9. *For any nontrivial property, there exist CA $\mathcal{A}_0$ and $\mathcal{A}_1$ over the same alphabet $Q_\mathcal{A}$ such that one among $\Lambda_\mu(\mathcal{A}_0)$ and $\Lambda_\mu(\mathcal{A}_1)$ has this property, and the other not.*

Proof. Consider some nontrivial property, there exist two CA $\mathcal{D}_O$ and $\mathcal{D}_1$ over alphabets Q_0 and Q_1, such that one among $\Lambda_\mu(\mathcal{D}_0)$ and $\Lambda_\mu(\mathcal{D}_1)$ has this property, and the other not. For simplicity, we suppose they both have radius 1, other cases could be treated the same way. We will take multiple copies of each state of each alphabet in order to obtain two CA with the same alphabet. Then, the new automata will consider a copy of a state as the equivalent of it. Copies will not appear after the initial configuration.

Suppose $|Q_0| = c$ and $|Q_1| = d$. We take the alphabet $Q = \{\alpha_{i_0}, \alpha_{i_1}, \ldots, \alpha_{i_e}\}$ with $e = \text{lcm}(c, d)$ and $Q_0 \cup Q_1 \subseteq Q$. Then, we partition twice Q into c sets $\{S_0(\gamma), \gamma \in Q_0\}$ with $|S_0(\gamma)| = |S_0(\gamma')|$, and into d sets $\{S_1(\gamma), \gamma \in Q_1\}$ with $|S_1(\gamma)| = |S_1(\gamma')|$. Then let $\mathcal{A}_0$ and $\mathcal{A}_1$ be CA over Q, of radius 1 and with the following rules:

- $\delta_{\mathcal{A}_0}(\alpha_x, \alpha_y, \alpha_z) = \alpha_t$ if $\delta_{\mathcal{D}_0}(\alpha_{x'}, \alpha_{y'}, \alpha_{z'}) = \alpha_t$ with $\alpha_x \in S_0(\alpha_{x'})$, $\alpha_y \in S_0(\alpha_{y'})$ and $\alpha_z \in S_0(\alpha_{z'})$,
- $\delta_{\mathcal{A}_1}(\alpha_x, \alpha_y, \alpha_z) = \alpha_t$ if $\delta_{\mathcal{D}_1}(\alpha_{x'}, \alpha_{y'}, \alpha_{z'}) = \alpha_t$ with $\alpha_x \in S_1(\alpha_{x'})$, $\alpha_y \in S_1(\alpha_{y'})$ and $\alpha_z \in S_1(\alpha_{z'})$,

Then:

- $\forall n \in \mathbb{N}, \forall u \in Q_0^*, {\mathcal{A}_0}^n\mu([u]) = {\mathcal{D}_0}^n\mu([u])$,
- $\forall n \in \mathbb{N}, \forall u \notin Q_0^*, {\mathcal{A}_0}^n\mu([u]) = 0$,
- $\forall n \in \mathbb{N}, \forall u \in Q_1^*, {\mathcal{A}_1}^n\mu([u]) = {\mathcal{D}_1}^n\mu([u])$,
- $\forall n \in \mathbb{N}, \forall u \notin Q_1^*, {\mathcal{A}_1}^n\mu([u]) = 0$.

□

Theorem 2. *Any nontrivial property of μ-limit sets of cellular automata is undecidable.*

Proof. We consider a nontrivial property $\mathcal{P}$ over μ-limit sets. There exist CA $\mathcal{A}_0$ and $\mathcal{A}_1$ such that $\Lambda_\mu(\mathcal{A}_0) \in \mathcal{P}$ and $\Lambda_\mu(\mathcal{A}_1) \notin \mathcal{P}$. With a CA $\mathcal{H}$ containing a permanent state, we construct two CA $\mathcal{B}_i$ ($i = 0$ or 1) with the construction described in 5.1, such that:

1. if $\mathcal{H}$ is μ-nilpotent then $\Lambda_\mu(\mathcal{B}_i) = \Lambda_\mu(\mathcal{A}_i)\}$ for $i = 0, 1$.
2. if $\mathcal{H}$ is not μ-nilpotent then $\Lambda_\mu(\mathcal{B}_0) = \Lambda_\mu(\mathcal{B}_1)$.

If there exists an algorithm that determines if the μ-limit set of a given CA has property $\mathcal{P}$, we apply it on CA $\mathcal{B}_0$ and $\mathcal{B}_1$. If the algorithm gives different answers for both CA, necessarily we have case 1 and $\mathcal{H}$ is μ-nilpotent. If on the other hand, both answers are identic, we have case 2 and $\mathcal{H}$ is not μ-nilpotent. Therefore, we have an algorithm to determine if a CA is μ-nilpotent. Which is a contradiction with [BPT06], where this problem was proved undecidable.

□