Everywhere Zero Pointwise Lyapunov Exponents for Sensitive Cellular Automata

Toni Hotanen

Everywhere Zero Pointwise Lyapunov Exponents for Sensitive Cellular Automata

Toni Hotanen

University of Turku, Turku, Finland
tonhot@utu.fi

Abstract. Lyapunov exponents are an important concept in differentiable dynamical systems and they measure stability or sensitivity in the system. Their analogues for cellular automata were proposed by Shereshevsky and since then they have been further developed and studied. In this paper we focus on a conjecture claiming that there does not exist such a sensitive cellular automaton, that would have both the right and the left pointwise Lyapunov exponents taking the value zero, for each configuration. In this paper we prove this conjecture false by constructing such a cellular automaton, using aperiodic, complete Turing machines as a building block.

Keywords: cellular automata · sensitive · Lyapunov exponents

1 Introduction

Cellular automata are discrete dynamical systems, consisting of a set of configurations over a regular lattice of cells, a finite set of symbols, a neighbourhood vector of cells and a local update rule. The local update rule, together with the neighbourhood, defines a global rule. The global rule takes each cell to a state dictated by the local rule, when given the states of a given cell's neighbours as an input. The regular lattice we are interested in, is often a countable group. Classically the group is chosen to be $\mathbb{Z}^d$ for some $d \in \mathbb{Z}_+$, where d is called the dimension of the cellular automaton. The set of configurations is the set of all mappings from the selected group to the finite set of symbols. The global function is continuous with respect to the prodiscrete topology equipped to the set of configurations.

Lyapunov exponents are a measure for the rate of divergence of infinitesimally close trajectories in dynamical systems. They were first introduced in 1892 by Lyapunov in his doctoral thesis titled: The general problem of the stability of motion, English translation of which can be found in [11]. Since then they have been widely studied in the context of differentiable dynamical systems. Their importance in the study of non-linear dynamical systems, for example, can be found stated in [6]. In the context of cellular automata, the Lyapunov exponents were first considered by Wolfram in [14]. In Problem 2 of [15], the question to establish the exact connection between entropies and Lyapunov exponents is asked. A first formal definition of Lyapunov exponents for one-dimensional cellular automata is due to Shereshevsky in [12], which he defines as a shift-invariant measure of left and right perturbation speeds. In fact he proves an inequality connecting the measure-theoretical entropy of a cellular automaton and its shift-invariant Lyapunov exponents. Tisseur altered the definitions slightly and considers average Lyapunov exponents in [13], where he establishes a similar connection. Finally the pointwise Lyapunov exponents are defined in [2].

Connections between various dynamical properties of cellular automata and possible values of Lyapunov exponents have been studied for example in [5], [4] and [2]. In [4], a closed formula for calculating the value of the shift-invariant Lyapunov exponents, was presented in the setting of linear cellular automata. In [5], the authors proved that for a given positively expansive cellular automaton, the shift-invariant Lyapunov exponents are positive for each configuration. The result was later improved in [2], where it was shown that the same result holds for the pointwise Lyapunov exponents.

In [2], Bressaud and Tisseur construct a sensitive cellular automaton, for which the value of the left and right average Lyapunov exponents is zero, with respect to a specific measure. In the same paper the Conjecture 3 states, that for a given sensitive cellular automaton it is necessary, that there exists such a configuration, whose either left or right pointwise Lyapunov exponent has a positive value. The same conjecture is reinstated by Kůrka as a Conjecture 11 in [9]. Our aim is to prove this conjecture false, which we will do in Theorem 2. The result follows from the existence of complete, aperiodic Turing machines, established in [1] and [3], and their movement bounds proved in [8] and [7].

2 Preliminaries

An *alphabet* Σ is a finite set of *symbols*. A *word* of length n over an alphabet Σ is any element $w = (w_0, w_1, \dots w_{n-1}) = w_0 w_1 \cdots w_{n-1}$ from the set $\Sigma^{[0,n)} = \Sigma^n$ and $|w| = n$ is the *length* of a word w. The *empty word* is denoted as ϵ and it is the unique word of length zero. A set of all finite words i.e. $\bigcup_{n \in \mathbb{N}} \Sigma^n$ is denoted as Σ^* and a set of all finite non-empty words $\Sigma^* \setminus \{\epsilon\}$ is denoted as Σ^+. A *concatenation* $\cdot : (\Sigma^*)^2 \to \Sigma^*$ is a mapping such that $u \cdot v = u_0 u_1 \dots u_n v_0 v_1 \dots v_m$, where $u = u_0 u_1 \dots u_n$ and $v = v_0 v_1 \dots v_m$. We will adapt the shorthand notation uv for the concatenation of any two words. Elements from the sets $\Sigma^{\mathbb{N}}$, $\Sigma^{\mathbb{Z}_-}$ and $\Sigma^{\mathbb{Z}}$ are called *right-infinite*, *left-infinite* and *bi-infinite* words, respectively. Furthermore we define a set $\Sigma^{\Omega} = \Sigma^+ \cup \Sigma^{\mathbb{N}} \cup \Sigma^{\mathbb{Z}_-} \cup \Sigma^{\mathbb{Z}}$. A concatenation of elements $u \in \Sigma^{\Omega}$ and $v \in \Sigma^{\Omega}$ is defined when u is finite or left-infinite and v is finite or right-infinite. Let $u \in \Sigma^{\Omega}$ and $w \in \Sigma^{\Omega}$, we will denote $u \sqsubset w$ if there exists such $j \in \mathbb{Z}$, that $u_{i+j} = w_i$ for each i in the domain of u, and say that u is a *subword* of w. If Σ and Γ are two alphabets, we will denote the set $\{uv \mid u \in \Sigma^{\alpha}, v \in \Gamma^{\beta}\}$ as $\Sigma^{\alpha}\Gamma^{\beta}$. where $\alpha \in \{\mathbb{Z}_-, *, +\} \cup \mathbb{N}$ and $\beta \in \{\mathbb{N}, *, +\} \cup \mathbb{N}$. In this notation, if $\Sigma = \{a\}$, we will omit the brackets. Finally if $w \in \Sigma^*$, we will use the notation w^{∞} for the right-infinite word $ww \cdots$.

A *Turing machine* is a 3-tuple (Q, Γ, δ), where Q is a finite set of *states*, Γ is a finite set of *symbols* and δ is a partial mapping $\delta : Q \times \Gamma \to Q \times \Gamma \times \Delta$ called a *transition rule*, where $\Delta = \{\leftarrow, \rightarrow\}$. A *configuration* is a 3-tuple (w, i, q), where $w \in \Gamma^{\mathbb{Z}}$, $i \in \mathbb{Z}$ and $q \in Q$. We will write $(w, i, q) \vdash (w', j, r)$ if $\delta(q, w_i) = (r, w'_i, d)$, where $w' \in \Gamma^{\mathbb{Z}}$, and $j = i+1$ if $d = \rightarrow$ and $j = i-1$ if $d = \leftarrow$ and $w'_k = w_k$ for each $k \neq i$. Inductively we define $\vdash^n$, where $\vdash$ is applied n times. Furthermore we will write $(w, i, q) \vdash^+ (w', j, r)$ if there exists such $n \in \mathbb{Z}_+$, that $(w, i, q) \vdash^n (w', j, r)$ holds. We will call a pair $(q, a) \in Q \times \Gamma$ an *error pair* if $\delta(q, a)$ is undefined. A Turing machine is called *complete* if it has no error pairs. We will call a configuration (w, i, q) *periodic* if $(w, i, q) \vdash^+ (w, i, q)$ and *weakly periodic* if there exists such $j \in \mathbb{Z}$, that $(w, i, q) \vdash^+ (w', i+j, q)$, where $w'_{k+j} = w_k$, for each $k \in \mathbb{Z}$. We will call a Turing machine *periodic* if all its configurations are periodic, and *aperiodic* if none of its configurations are weakly periodic.

A *topological dynamical system* is a pair (X, f), where X is a compact metric space and f is a continuous function $f\colon X \to X$.

A *shift dynamical system* is a dynamical system $(\Sigma^{\mathbb{Z}}, \sigma)$, where Σ is a finite set of symbols, $\Sigma^{\mathbb{Z}}$ is the space called the *full shift* and σ, called the *shift*, is defined in a way that $\sigma(x)_i = x_{i+1}$. The *metric* d of the space $\Sigma^{\mathbb{Z}}$ is defined as $d(x, y) = 2^{-\inf\{|i| \in \mathbb{N} | x_i \neq y_i\}}$. It is not difficult to see that the space $\Sigma^{\mathbb{Z}}$ is compact and that the function σ is continuous. An *endomorphism* is a continuous function $f\colon \Sigma^{\mathbb{Z}} \to \Sigma^{\mathbb{Z}}$, such that $f \circ \sigma = \sigma \circ f$.

A *one-dimensional cellular automaton* is a 3-tuple $\mathcal{A} = (\Sigma, N, h)$, where Σ is a finite set of symbols called *states*, N is a *neighbourhood* $(i_1, i_2, \ldots, i_n) \in \mathbb{Z}^n$ and $h : \Sigma^n \to \Sigma$ is a *local rule*. If $N = [-r, r]$, we call N a *radius-r* neighbourhood. In the context of cellular automata, we call the full shift $\Sigma^{\mathbb{Z}}$ a *configuration space* and refer to its elements as *configurations*. The local rule together with the neighbourhood induces a global rule $f : \Sigma^{\mathbb{Z}} \to \Sigma^{\mathbb{Z}}$, which is defined in such a way that $f(c)_i = h(c_{i+i_1}, c_{i+i_2}, \cdots, c_{i+i_n})$. We make no distinction between a cellular automaton and its global rule. By the Curtis-Hedlund-Lyndon theorem, the cellular automata, sometimes abbreviated as CA, are exactly the endomorphisms of the shift dynamical systems.

Definition 1. *A dynamical system (X, f) is sensitive if*

$$\exists \epsilon > 0\colon \forall \delta > 0\colon \forall x \in X\colon \exists y \in B_\delta(x)\colon \exists n \in \mathbb{N}\colon\ f^n(y) \notin B_\epsilon(f^n(x)).$$

Definition 2. *Let (Σ, N, h) be a one-dimensional cellular automaton, with a global rule $f : \Sigma^{\mathbb{Z}} \to \Sigma^{\mathbb{Z}}$. For every $c \in \Sigma^{\mathbb{Z}}$, we define*

$$W_m^+(c) = \{c' \in \Sigma^{\mathbb{Z}} \mid \forall i \geq m, c'_i = c_i\}$$

and

$$W_m^-(c) = \{c' \in \Sigma^{\mathbb{Z}} \mid \forall i \leq m, c'_i = c_i\}.$$

Furthermore we define

$$I_n^+(c) = \min\{m \in \mathbb{N} \mid f^i(W_{-m}^+(c)) \subseteq W_0^+(f^i(c)), \forall i \leq n\}$$

and

$$I_n^-(c) = \min\{m \in \mathbb{N} \mid f^i(W_m^-(c)) \subseteq W_0^-(f^i(c)), \forall i \leq n\}$$

Finally we define the pointwise Lyapunov exponents as

$$\lambda^+(c) = \liminf_{n\to\infty} \frac{I_n^+(c)}{n}$$

and

$$\lambda^-(c) = \liminf_{n\to\infty} \frac{I_n^-(c)}{n}.$$

Definition 3. *Let X be a set. A relation is a subset $R \subseteq X \times X$. We will use the standard notation aRb if $(a,b) \in R$. We will denote the complement of R as R^c, i.e. $R^c = (X \times X) \setminus R$.*

One way to simulate Turing machines in the CA setting, is to use the moving head model, or TMH for short, as introduced in [10] and by adding arrows, which separate different simulation areas from each other. We will be using a slight variation.

In the following definition we will describe a function induced from a given Turing machine. The function can then be used to define different types of local rules for CA. The construction is using the conveyor belt model, which has been used previously at least in [7], albeit the notations might differ.

Definition 4. *Let $\mathcal{M} = (Q, \Gamma, \delta)$ be a Turing machine. We will define an alphabet for the conveyor belt model. Elements from Q_2 will be called Turing machine heads and elements from T_2 will be called tape symbols. The set $\Delta = \{\rightarrow, \leftarrow\}$, consisting of direction symbols, is used for locating the Turing machine heads. The conveyor belt model uses two layers of tape, where the second one is reversed and the two tapes are connected at the ends of the simulation areas. We will define a relation, which allows us to define these simulation areas as simulation words, each consisting of at most one Turing machine head and where directions are forced to point to the unique Turing machine head. Let $\Gamma_2' = Q_2 \cup T_2$, where $Q_2 = \Gamma^2 \times Q \cup Q \times \Gamma^2$, $T_2 = \Gamma^2 \times \Delta$.*

Define a relation R_2 in a following way: Let $a \in \Gamma_2'$ and $b \in \Gamma_2'$, then

$$aR_2b \text{ if } \begin{cases} & a \in \Gamma^2 \times \{\rightarrow\} \wedge b \in (\Gamma^2 \times \{\rightarrow\}) \cup Q_2 \\ \vee & a \in Q_2 \wedge b \in \Gamma^2 \times \{\leftarrow\} \\ \vee & a \in \Gamma^2 \times \{\leftarrow\} \wedge b \in \Gamma^2 \times \{\leftarrow\}. \end{cases}$$

Now we can define a set of simulation words:

$$\Sigma_{\mathcal{M},2} = \{w \in \Gamma_2'^{\Omega} \mid w_j R_2 w_{j+1}, \forall j\}.$$

In the following, in the case that the input word contains a Turing machine head, we will assume $uvwxy$ to be such a decomposition of the input word, that $w \in Q_2$ and that the following conditionals hold: If $|uv| > 0$, then $|v| = 1$. If $|xy| > 0$, then $|x| = 1$. This decomposition is then unique. In the cases when the input word consists of only tape symbols the decomposition can be arbitrary.

From the transition rule of the Turing machine, we can construct a mapping $m_2 : \Sigma_{\mathcal{M},2} \to \Sigma_{\mathcal{M},2}$ *in the following way:*

$$m_2(uvwxy) = \begin{cases} uv'w'xy & \text{if } w = (q,a,a') \wedge v = (b,b',\to) \wedge \delta(q,a) = (r,c,\leftarrow), \\ & \text{where } v' = (r,b,b') \wedge w' = (c,a',\leftarrow), \\ w'xy & \text{if } uv = \epsilon \wedge, w = (q,a,a') \wedge \delta(q,a) = (r,c,\leftarrow), \\ & \text{where } w' = (c,a',r), \\ uv'w'xy & \text{if } w = (a,a',q) \wedge v = (b,b',\to) \wedge \delta(q,a') = (r,c',\to), \\ & \text{where } v' = (b,b',r) \wedge w' = (a,c',\leftarrow), \\ w'xy & \text{if } uv = \epsilon \wedge, w = (a,a',q) \wedge \delta(q,a') = (r,c',\to), \\ & \text{where } w' = (r,a,c'), \\ uvw'x'y & \text{if } w = (q,a,a') \wedge x = (b,b',\leftarrow) \wedge \delta(q,a) = (r,c,\to), \\ & \text{where } x' = (r,b,b') \wedge w' = (c,a',\to), \\ uvw' & \text{if } xy = \epsilon \wedge, w = (q,a,a') \wedge \delta(q,a) = (r,c,\to), \\ & \text{where } w' = (c,a',r), \\ uvw'x'y & \text{if } w = (a,a',q) \wedge x = (b,b',\to) \wedge \delta(q,a') = (r,c',\leftarrow), \\ & \text{where } x' = (b,b',r) \wedge w' = (a,c',\to), \\ uvw' & \text{if } xy = \epsilon \wedge, w = (a,a',q) \wedge \delta(q,a') = (r,c',\leftarrow), \\ & \text{where } w' = (r,a,c'), \\ uvwxy & \text{otherwise.} \end{cases}$$

Definition 5. *Let* $\mathcal{M} = (Q, \Gamma, \delta)$ *be a Turing machine. Using the notations introduced in Definition 4, we define a tracking function* $T : \Sigma_{\mathcal{M},2} \times \mathbb{N} \to \mathbb{Z}$*, where* $T(w,j) = k$ *if* $m_2^j(w)_k \in Q_2$ *and* $T(w,j) = 0$ *if* $w_i \notin Q_2$ *for each* i *in the inputs domain. We also define the number of indices visited by a Turing machine head in* j *steps, given some initial word, by a function* $V(w,j) : \Sigma_{\mathcal{M},2} \times \mathbb{N} \to \mathbb{N}$*, defined as* $V(w,j) = |\{T(w,j') \in \mathbb{Z} \mid j' \leq j\}|$*. Finally we define a movement bound* $M : \mathbb{N} \to \mathbb{N}$ *as a function such that* $M(j) = max_{w \in \Sigma_{\mathcal{M},2}} V(w,j)$*.*

The sublinearily of the movement bound for aperiodic Turing machines is already proved in [8]. We will however make use of the following tighter bound.

Theorem 1. *[7] Let* $\mathcal{M} = (Q, \Gamma, \delta)$ *be Turing machine, and* M *its movement bound. If* $\mathcal{M}$ *is aperiodic, then* $M = \mathcal{O}(\frac{n}{\log n})$*.*

3 Lyapunov exponents for sensitive cellular automata

In this section we study the question whether or not the sum of the pointwise Lyapunov exponents of a given sensitive CA can take the value 0 or not. We use the notations of the previous sections without explicitly referring to them. In Theorem 2 we show how to construct a sensitive cellular automaton for a given aperiodic Turing machine, such that its left and right pointwise Lyapunov exponents are bounded from above by a sublinear function derived from the movement bound of the Turing machine.

Theorem 2. *There exists such a sensitive one-dimensional cellular automaton* (Σ, N, f)*, that* $\lambda^+(c) = \lambda^-(c) = 0$ *for every configuration* $c \in \Sigma^{\mathbb{Z}}$*.*

Proof. Let $\mathcal{M} = (Q, \Gamma, \delta)$ be an aperiodic, complete Turing machine. We will alter $\mathcal{M}$ slightly to produce a new Turing machine $\mathcal{M}' = (Q', \Gamma, \delta')$, where $Q' = Q \times \{0, 1\}$ and $\delta'((q, x), a) = ((r, x), b, d)$, where $\delta(q, a) = (r, b, d)$. Essentially this modification gives us two copies of the original Turing machine and the computations of the new machine can be projected to the computations of the original machine. Using the new Turing machine, we can set our symbol set for the CA, which we are constructing, as $\Sigma = \Gamma'_2 \cup \{>\}$. We will refer the symbol $>$ as the *eraser*. We will also use the notations $Q_2^i = \Gamma^2 \times Q \times \{i\} \cup Q \times \{i\} \times \Gamma^2$, where $i \in \{0, 1\}$ and $Q_2 = Q_2^0 \cup Q_2^1$. For each configuration $c \in \Sigma^{\mathbb{Z}}$, we define a set of locations for the Turing machine heads as

$$H_c = \{i \in \mathbb{Z} \mid c_i \in Q_2\}$$

and a set of locations for the erasers as

$$E_c = \{i \in \mathbb{Z} \mid c_i = >\}.$$

Next we define the *simulation bounds* as functions $l_c : H_c \to \mathbb{Z} \cup \{-\infty\}$ and $r_c : H_c \to \mathbb{Z} \cup \{\infty\}$ in the following way:

$$l_c(i) = \sup\{j \in \mathbb{Z} \mid j \leq i \text{ and } c_{j-1} R_2^c c_j\}$$

and

$$r_c(i) = \inf\{j \in \mathbb{Z} \mid i \leq j \text{ and } c_j R_2^c c_{j+1}\}.$$

From these bounds we can define the set of cells that are not part of any simulation area as

$$U_c = \mathbb{Z} \setminus (\bigcup_{i \in H_c} (l_c(i) - 1, r_c(i) + 1)).$$

Using the simulation bounds, we can define a function, which simulates all Turing machines in their respective simulation areas as $m : \Sigma^{\mathbb{Z}} \to \Sigma^{\mathbb{Z}}$, where

$$\begin{aligned} m(c)_{(l_c(i)-1, r_c(i)+1)} &= m_2(c_{(l_c(i)-1, r_c(i)+1)}) \ \forall i \in H_c \text{ and} \\ m(c)_i &= c_i \ \forall i \in U_c. \end{aligned}$$

Clearly m is a cellular automaton since we can extract a radius-1 local rule from its definition.

Next we fix a state $a_0 \in \Gamma^2 \times \{\leftarrow\}$ and define a second cellular automaton $e : \Sigma^{\mathbb{Z}} \to \Sigma^{\mathbb{Z}}$, whose local rule is described in Table 1. The dynamics of e can be described in the following way: If a cell i, containing the eraser state $>$, sees a Turing machine head from the state set Q_2^1 at a cell $i - 1$, then the eraser state gets pushed to the next cell at $i + 1$. This happens regardless of the previous content of the cell $i + 1$. As a consequence, the simulation area in the left side increases by one cell. If there is a simulation area on the right side, then either its size is reduced by one or it is removed entirely. The simulation area gets removed if there is a Turing machine head at a cell $i+1$ or $i+2$. In such situation, the Turing machine head is replaced with a tape symbol a_0. Otherwise if a cell at state $>$ does not see a Turing machine head from the state set Q_2^1 to their immediate left, then it stays at its current state. When a Turing machine head from the state set Q_2^1 moves an eraser state, it changes to an equivalent symbol from the state set Q_2^0. If a Turing machine head from the state set Q_2^0 visits the left bound of the simulation area, which it belongs to, then if it does not get erased, it changes to an equivalent symbol from the state set Q_2^1. The idea is that the simulation areas can increase in their size arbitrarily far to the right, but only one cell at a time and in between the increments, the Turing machine head must visit the left bound of the simulation area.

<table>
<tr><td colspan="3">$q, q' \in Q_2, q_i = (q, i), x \in \Sigma$
$u \in \Sigma \setminus Q_2^1, u' \in \Sigma \setminus Q_2, v, v' \in \Sigma \setminus \{>\}, v'' \in \Gamma^2 \times \{\leftarrow\} \cup Q_2$</td></tr>
<tr><td>$q_1 \; > \; x \; u'$
$a_0 > u'$</td><td>$q_1 \; > \; x \; q'$
$a_0 > a_0$</td><td>$v \; v' \; q_1 \; >$
$q_0 \; a_0$</td></tr>
<tr><td>$u > u'$
$> u'$</td><td>$u > q'$
$> a_0$</td><td>$v \; v'' \; q_0$
q_1</td></tr>
</table>

Table 1. The definition of the local rule of the CA $e : \Sigma^{\mathbb{Z}} \to \Sigma^{\mathbb{Z}}$. Here a_0 is a fixed state from the set $\Gamma^2 \times \{\leftarrow\}$. The rule uses a neighbourhood $(-2, -1, 0, 1)$. Input is written on the first row and output on the second row. If the output is written, but the input is partially missing, it means that the missing cells do not affect the output. For the inputs that do not appear in the table, the local rule behaves as the identity mapping.

We are ready to define our cellular automaton of interest as $f : \Sigma^{\mathbb{Z}} \to \Sigma^{\mathbb{Z}}$, where $f = m \circ e$. The behaviour of the CA is depicted in Figure 1.

In Lemmas 1 and 2 we prove, that the CA we constructed has the desired properties and therefore the claim follows.

$\square$

Lemma 1. *The cellular automaton (Σ, N, f) constructed in Theorem 2 has the property that $\lambda^+(c) = \lambda^-(c) = 0$ for every configuration $c \in \Sigma^{\mathbb{Z}}$.*

Proof. We will begin the proof by introducing tracking functions for the eraser states, the simulation bounds and the Turing machine heads. The point of them is, that given an initial configuration and a cell containing a Turing machine head, an eraser or a simulation bound, we can tell to which cell said symbol has travelled to in time. In the output of each function, we will use the symbol $-$ to denote, that the symbol no longer exists, i.e. it has been destroyed by a symbol $>$. We will then show that a difference can only propagate inside each simulation area or by the movement of the simulation areas. In either case the movement is bounded from above by a sublinear function derived from the movement bound of the Turing machine.

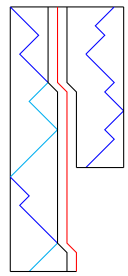

Fig. 1. In this figure, we have depicted the behaviour of the CA constructed in Theorem 2. The black lines represent the left and right simulation bounds, the blue and cyan lines represent Turing machine heads from the sets Q_2^1 and Q_2^0, respectively, and the red lines represent the erasers. We also note, that in the figure, time increases from top to bottom. One can witness several types of behaviour in the two simulation areas. When the Turing machine head in the state from the set Q_2^1, of the left simulation area, visits the right boundary, then in the next time-step, the eraser moves one cell to the right, which also moves the left boundary of the second simulation area. In the same time step, the Turing machine head switches to some state in the set Q_2^0. Such Turing machine heads do not move the eraser states as witnessed when the cyan coloured line visits the right boundary. For the Turing machine to be allowed to move the eraser state again, it needs to switch back to an element from Q_2^1, which happens if and only if it visits the left boundary. The right simulation area does not have an eraser on the right side and hence its simulation area can never increase in size. In the middle of the image we can see that the Turing machine head on the right simulation area visits a cell within the distance of two, to a cell containing an eraser state and hence gets removed. This happens eventually in all simulation areas, in which at some time-step an eraser state can be seen in a cell within distance two to its left boundary.

First we will define the tracking function for the eraser states as $e_c : E_c \times \mathbb{N} \to \mathbb{Z} \cup \{-\}$, where

$$e_c(i,j) = \begin{cases} i & \text{if } j = 0, \\ e_c(i,j-1)+1 & \text{if } f^{j-1}(c)_{[e_c(i,j-1)-1, e_c(i,j-1)]} \in Q_2^1 >, \\ - & \text{if } f^{j-1}(c)_{[e_c(i,j-1)-2, e_c(i,j-1)-1]} \in Q_2^1 >, \\ e_c(i,j-1) & \text{otherwise.} \end{cases}$$

Next we will define the tracking functions for the simulation bounds and for the Turing machine head, inductively with respect to $j \in \mathbb{N}$, as $l_c : H_c \times \mathbb{N} \to \mathbb{Z} \cup \{-\}$, $r_c : H_c \times \mathbb{N} \to \mathbb{Z} \cup \{-\}$ and $h_c : H_c \times \mathbb{N} \to \mathbb{Z} \cup \{-\}$, where

$$l_c(i,j)=\begin{cases} l_c(i) & \text{if } j=0,\\ l_c(i,j-1)+1 & \text{if } f^{j-1}(c)_{[l_c(i,j-1)-2,l_c(i,j-1)+1]}\in Q_2^1>T_2T_2,\\ l_c(i,j-1) & \text{if } f^{j-1}(c)_{[l_c(i,j-1)-2,l_c(i,j-1)-1]}\notin Q_2^1>,\\ & \text{and } >\not\sqsubset f^{j-1}(c)_{[h_c(i,j-1)-2,h_c(i,j-1)-1]},\\ - & \text{if } >\sqsubset f^{j-1}(c)_{[h_c(i,j-1)-2,h_c(i,j-1)-1]}, \end{cases}$$

$$r_c(i,j)=\begin{cases} r_c(i) & \text{if } j=0,\\ r_c(i,j-1)+1 & \text{if } l_c(i,j)\neq -\\ & \text{and } f^{j-1}(c)_{[r_c(i,j-1),r_c(i,j-1)+1]}\in Q_2^1>,\\ r_c(i,j-1) & \text{if } l_c(i,j)\neq -\\ & \text{and } f^{j-1}(c)_{[r_c(i,j-1),r_c(i,j-1)+1]}\notin Q_2^1>,\\ - & \text{if } l_c(i,j)=-, \end{cases}$$

and

$$h_c(i,j)=\begin{cases} i & \text{if } j=0,\\ k & \text{if } l_c(i,j)\leq k\leq r_c(i,j)\\ & \text{and } f^j(c)_k\in Q_2,\\ - & \text{if } l_c(i,j)=-. \end{cases}$$

We will show that for each $i'\in E_{f(c)}$ there exists unique $i\in E_c$, such that $e_c(i,1)=i'$. By the definition of f it follows immediately, that $c_{i'}=>$ or $c_{i'-1}=>$ and furthermore if only the latter holds, then necessarily $c_{i'-2}\in Q_2^1$. By the definition of e_c if $c_{[i'-2,i'-1]}\in Q_2^1>$, then $i'=i+1$ and otherwise $i=i'$. Then furthermore by a straightforward induction, it holds that for each $n\in\mathbb{N}$ and $i'\in E_{f^n(c)}$ there exists unique $i\in E_c$, such that $e_c(i,n)=i'$.

Analogously we will show that for each $i'\in H_{f(c)}$ there exists unique $i\in H_c$, such that $h_c(i,1)=i'$, $l_c(i,1)=l_{f(c)}(i')$ and $r_c(i,1)=r_{f(c)}(i')$. From the definition of the CA m, it follows immediately that there exists such $i\in[i'-1,i'+1]$, that $i\in H_c$. Let us first assume, that such i is unique. Then by definition $l_c(i)=\sup\{j\in\mathbb{Z}\mid j\leq i \text{ and } c_{j-1}R_2^c c_j\}$. Clearly if $l_c(i)=-\infty$, then also $l_{f(c)}(i')=-\infty$. Let us then assume that $l_c(i)\in\mathbb{Z}$. From the definition of m and e it follows that $l_{f(c)}(i')\in[l_c,l_c+1]$ and furthermore $l_{f(c)}(i')=l_c+1$ if and only if $c_{[l_c-2,l_c-1]}\in Q_2^1>$. But this is consistent with the definition of $l_c(i,1)$ and hence $l_c(i,1)=l_{f(c)}(i')$. Similarly we can show that $r_c(i,1)=r_{f(c)}(i')$. It is then apparent, that $h_c(i,1)=i'$ as cell i' is the only cell containing a Turing machine head in the interval $(l_c(i,1)-1,r_c(i,1)+1)$. If there exist multiple Turing machine heads in the interval $[i'-1,i'+1]$, then let us first assume that $i'-1\in H_c$ and $i'+1\in H_c$. Then if $i'\in H_c$ it follows that $l_c(i')=i'=r_c(i')$ and since $i'-1\neq>$ and $i'+1\neq>$, we have that $l_c(i',1)=i'=r_c(i',1)$ and hence the claim follows. If $i'\notin H_c$, then $i'\in T_2$, hence either we have that $r_c(i'-1)=i'$

and $l_c(i'+1) = i'+1$ or $l_c(i'+1) = i'$ and $r_c(i'-1) = i'-1$. We only prove the case of the former as the latter is analogous. Immediately it follows that $r_c(i'-1,1) = i' = r_{f(c)}(i')$ as $i'+1 \neq >$. The analysis that results in to showing that $l_c(i'-1,1) = l_{f(c)}(i')$ is similar to the case where there was only one Turing machine head in the interval $[i'-1, i'+1]$. Hence we have that $h_c(i,1) = i'$, where $i = i'-1$. The case when $i'-1 \in H_c$, $i' \in H_c$, but $i'+1 \notin H_c$ and the case when $i'-1 \notin H_c$, $i' \in H_c$, but $i'+1 \in H_c$ are proved similarly. Now again by a straightforward induction it can be proved that for each $n \in \mathbb{N}$ and $i' \in H_{f^n(c)}$ there exists unique $i \in H_c$, such that $l_c(i,n) = l_{f^n(c)}(i')$, $r_c(i,n) = r_{f^n(c)}(i')$ and $h_c(i,n) = i'$.

Denote by $S_c(i,j) = (l_c(i,j)-1, r_c(i,j)+1)$, $w_{i,j} = c_{S_c(i,j)}$ and $s_{i,j} = |w_{i,j}|$. Suppose that $i \in H_c$ and $j \in \mathbb{N}$ are such that $r_c(i,j+1) = r_c(i,j)+1$. By definition of the tracking function, we have that $f^j(c)_{[r_c(i,j), r_c(i,j)+1]} \in Q_2^1 >$ and by the definition of the CA e, we have that $f^{j+1}(c)_{h_c(i,j+1)} \in Q_2^0$. We assume that there exists such $k \in \mathbb{N}$, that $r_c(i,j+1) < r_c(i,j+1+k)$. Then necessarily, from the aperiodicity of the Turing machine, we get that there exists a minimal such $k' \geq 1$, that $h_c(i,j+k') = l_c(i,j+k')$. From the definition of e, this means that $f^{j+k''}(c)_{h_c(i,j+k'')} \in Q_2^0$ for each $1 \leq k'' \leq k'$ and $f^{j+k'+1}(c)_{h_c(i,j+k'+1)} \in Q_2^1$. Notice that it also implicitly holds that $> \not\sqsubset f^t(c)_{[l_c(i,t)-2, l_c(i,t)-1]}$ for each $t \leq j+k'$ as otherwise we would have $h_c(i,j+k') = -$. We denote by $M^{-1}(n) = \min\{t \in \mathbb{N} \mid M(t) \geq n\}$ for each $n \in \mathbb{N}$. By the movement bound of the Turing machine we have that $M^{-1}(s_{i,j}) \leq k' \leq p^{s_{i,j+1}}$, where $p = |\Sigma|$. Furthermore from the definition of k, we have that $M^{-1}(s_{i,j+1}) \leq k - k' \leq p^{s_{i,j+1}}$ and hence $2M^{-1}(s_{i,j}) \leq k \leq 2p^{s_{i,j+1}}$. It is easy to see that during these k steps a valid Turing machine computation is performed with the input word $w_{i,j+k}$. Finally let $z_j = s_{i,j} - s_{i,0}$ and assume that $r_c(i,j) = h_c(i,j)$. From the above considerations we get that $M^{-1}(s_{i,0}+z_j-1) \leq j \leq 2\sum_{1 \leq k \leq z_j} p^{(s_{i,0}+k)} \leq p^{(s_{i,0}+2+z_j)}$. From the lower bound it follows that $|\{\alpha_{i,j} \in \mathbb{N} \mid j \leq n\}| \leq M(n)$, for each $n \in \mathbb{N}$ and $i \in H_c \cup E_c$, where α is any of the tracking functions for either the simulation bounds, the Turing machine heads or the erasers.

We are ready to prove the claim of the theorem. Let $c \in \Sigma^{\mathbb{Z}}$. For an infinite set of integers $n \in \mathbb{N}$, we want to find such values $m_n \in \mathbb{N}$, that if $c' \in W^+_{-m_n}(c)$ and there exists j_n, such that $f^{j_n}(c') \notin W^+_{-m_n}(f^{j_n}(c))$, then $j_n > n$. For each n, we can first assume that $m_n \geq M(n)$. We will analyse three cases: 1) None of the cells in the interval $[-m_n-3, 0]$ are in the eraser state in either of the configurations, during the first n iterations. 2) There is an upper bound $k < n$, after which none of the cells in the interval $[-m_n-3, 0]$ are in the eraser state in either of the configurations. 3) One of the configurations has a cell in the interval $[-m_n-3, 0]$, which is at an eraser state at the n^{th} iteration.

First let us assume that $> \not\sqsubset f^j(c'')_{[-m_n-3,0]}$ holds for each $j \leq n$ and $c'' \in \{c, c'\}$. Then by the definition of the global rule e, we have that $e(f^j(c''))_i = f^j(c'')_i$ for each $j \leq n$, $i \in [-m_n, 1]$ and $c'' \in \{c, c'\}$. Hence if $j_n \leq n$, is such that $f^{j_n}(c') \notin W^+_{-m_n}(f^{j_n}(c))$, then the difference must be caused by applying the global rule m. By the definition of m, this means that $f^{j_n}(c'')_0 \in Q_2$ for either $c'' = c$ or $c' = c$. From the movement bound and the assumption that

$m_n \geq M(n)$, it then follows that there exists such $i \in H_{c''}$, that $h_{c''}(i,j) \in [-m_n, m_n]$ for each $j \leq n$. Since $c' \in W^+_{-m_n}(c)$, then $i \in H_c \cap H_{c'}$ and hence locally within the cells in $[-m_n, m_n]$, the same Turing machine computation is simulated for the n steps and thus $j_n > n$.

Assume then that there exists such $j \leq n$ and $c'' \in \{c, c'\}$, that $> \sqsubset f^j(c'')_{[-m_n-3,0]}$. We further assume, that there exists minimal such $k < n$, that $> \not\sqsubset f^j(c'')_{[-m_n-3,0]}$ for each $c'' \in \{c, c'\}$ and $j \in \mathbb{N}$, such that $k < j \leq n$. Then due to definition of e, necessarily for either $c'' = c$ or $c'' = c'$ it holds that $f^k(c'')[-1,0] \in Q_2^1 >$ and $> \not\sqsubset f^k(c'')_{[-m_n-3,1]}$. We can assume that this holds for $c'' = c$. Then, from the movement bound it follows, that there exists such $i \in H_c$, that $h_c(i,j) \in [-m_n - 1, -1]$ for each $j \leq k$. We clearly have that $i \in H_{c'}$. If $h_{c'}(i,j) = h_c(i,j)$ for each $j \leq n$, then a difference cannot propagate to the origin during the n steps as again the same computation would happen in both configurations. On the other hand the only way that $h_{c'}(i,j) \neq h_c(i,j)$ for some $j \leq n$ is if $h_{c'}(i,j) = -$. This is however impossible as it would require that $> \sqsubset c'_{[-m_n-3,i-1]}$, which would mean that $> \sqsubset f^{k+1}(c')_{[-m_n-3,0]}$, which is against the assumption. Therefore again if $j_n \in \mathbb{N}$, is such that $f^{j_n}(c') \notin W^+_{-m_n}(f^{j_n}(c))$, then from the above consideration, we have that $j_n \geq n$.

Finally, for the last case we assume that $> \sqsubset f^n(c)_{[-m_n-3,0]}$. Then from the movement bound M, it follows that there exists such $i_e \in E_c$, that $e_c(i_e, j) \in [-2M(n) - 3, 0]$ for each $j \leq n$. We can then assume that $m_n \geq 2M(n) + 3$ and hence $i_e \in E_{c'}$. If $e_c(i_e, j) = e_{c'}(i_e, j)$ for each $j \leq n$, then directly from the definition of the CA f, it follows that $f^j(c)_k = f^j(c')_k$, for each $j \leq n$ and $k \geq e_c(i_e, j)$ and hence in such case $j_n > n$. Therefore we consider the case where there exists such $j \leq n$, that $e_{c'}(i_e, j) \neq e_c(i_e, j)$. Suppose that for each pair $i \in H_c$ and $i' \in E_c$, such that $i < i' = r_c(i) + 1 < i_e$, it holds that either there exists minimal such $j_0 \leq n$, that $e_{c''}(i', j_0) - 2 \geq h_{c''}(i, j_0)$ and $e_c(i_e, j) = e_c(i_e, j_0)$ for each $j_0 < j \leq n$ or $e_{c''}(i', j) - 2 < h_{c''}(i, j)$ for each $j \leq n$, where $c'' \in \Sigma^{\mathbb{Z}}$ is such a configuration that $c''_{j'} = c_{j'}$ for each $j' > l_c(i) - 1$ and $c''_{j'} = a_0$ for each $j' < l_c(i)$. If we now assume that $m_n \geq 3M(n) + 4$, then any $i'' \in E_{c'}$, such that $i'' < -m_n$ cannot effect the value of $e_{c'}(i_e, j)$ for any $j \leq n$ and neither can any $i'' \in E_{c'} \cap [-m_n, i_e - 1]$ by our assumption regarding configurations c''. Hence if $e_c(i_e, j) \neq e_{c'}(i_e, j)$ and $j \leq n$, the difference is caused by a Turing machine head. But during the first n steps said Turing machine head can only visit cells $[-M(n) + i_e - 1, i_e + M(n)] \subseteq [-3M(n) - 4, M(n)]$. So again both configurations c and c' are simulating the exact same computation during the first n steps and thus $j_n > n$.

Let us assume that there exists such $k \in \mathbb{N}$ and indices $i_j \in H_c$, where $1 \leq j \leq k + 1$, that $i_j < i_{j+1}$ for each $j \leq k$. Let us denote as $c^j \in \Sigma^{\mathbb{Z}}$ such a configuration that $c^j_{j'} = c_{j'}$ for each $j' > l_c(i_j) - 1$ and $c^j_{j'} = a_0$ for each $j' < l_c(i_j)$. We will also assume that for each $j \leq k + 1$, we have that $e_{i_j} = r_c(i_j) + 1 \in E_c$ and we will further assume that there exists such an increasing finite sequence of times t_j, that $h_{c^j}(i_{j+1}, t_j) \in [e_{c^j}(e_{i_j}, t_j) + 1, e_{c^j}(e_{i_j}, t_j) + 2]$ and $e_{c^{j+1}}(e_{i_{j+1}}, t_{j+1}) > e_{c^{j+1}}(e_{i_{j+1}}, t_j)$ for each $j \leq k$. That is the leftmost simulation area in c^j, destroys the simulation area of the Turing machine head i_{j+1} at time-step t_j. In c however, this might not happen as there could be another simulation, which destroys the leftmost simulation area of c^j before time t_j. This could allow an existence for an alternating chain of such simulation areas, where simulation areas of i_j are destroyed for each even j in configuration c and and odd j in configuration c'. We want to find an upper bound $a(n)$ for the maximum distance $e_{c^k}(e_{i_k}, t_{i_k}) - e_{c^1}(e_{i_1}, t_{i_1})$, assuming that $t_k \leq n$, as then we know how fast a difference can potentially propagate via such a chain of simulation areas. In such a case, suppose that we would have that $c'_{[i_e - a(n) - 2M(n) - 4, \infty)} = c_{[i_e - a(n) - 2M(n) - 4, \infty)}$, where $i_e = e_{i_{k+1}}$. Assume that there exist such $j_0 \in \mathbb{N}$, that $e_c(i_e, j_0) \neq e_{c'}(i_e, j_0)$, then if the difference is due to a chain that we have described above, it must be that in some of the simulation areas of such chain, a different computation is performed in the configurations c and c'. If $i_e - e_{c^k}(e_{i_k}, t_{i_k}) > M(n) + 2$, then we would have that $e_c(i_e, j) = e_{c'}(i_e, j)$ for each $j \leq n$. If $i_e - e_{c^k}(e_{i_k}, t_{i_k}) \leq M(n) + 2$, then $e_{c^1}(e_{i_1}, t_{i_1}) \geq i_e - 2 - M(n) - a(n)$, but then in all of the simulation areas the same computation is performed during n iterations and hence $e_c(i_e, j) = e_{c'}(i_e, j)$ for each $j \leq n$.

Let us assume that for each $j \leq k$, we have that $p^n \leq t_j \leq p^{n+1}$, where $p = |\Sigma|$ and $n \geq 5$. Let $b_j = e_{c^{j+1}}(e_{i_{j+1}}, t_{j+1}) - e_{c^j}(e_{i_j}, t_j)$ and $b_{j,t} = e_c(e_{i_{j+1}}, t) - e_c(e_{i_j}, t)$. Since $h_{c^j}(i_{j+1}, t_j) \in [e_{c^j}(e_{i_j}, t_j)+1, e_{c^j}(e_{i_j}, t_j)+2]$ and $e_{c^{j+1}}(e_{i_{j+1}}, t_{j+1}) > e_{c^{j+1}}(e_{i_{j+1}}, t_j)$ for each $j \leq k$, there must exist such $j' \leq t_{j+1} - t_j$, that $h_{c^{j+1}}(i_{j+1}, t_j + j') = e_{c^{j+1}}(e_{i_{j+1}}, t_{j+1}) - 1$ and therefore we have that $b_j - 2 \leq M(t_{j+1} - t_j)$ for each $j \leq k$.

By Theorem 1 there exists such $h : \mathbb{N} \to \mathbb{N}$, that $M(n) \leq h(n) = C\frac{n}{\log(n)}$ for each $n \in \mathbb{N}$ and where $C > 0$. It is easy to see that there exists such a positive real number C', that h is concave in the domain $[C', \infty)$. We will split the times t_j into two sets A and B, such that $j \in A$ if $t_{j+1} - t_j \geq C'$ and $t_j \in B$ otherwise. We have that

$$
\begin{aligned}
\sum_{j=1}^{k} b_j &\leq 2k + \sum_{j=1}^{k} M(t_{j+1} - t_j) \\
&\leq 2k + \sum_{j=1}^{k} h(t_{j+1} - t_j) \\
&\leq 2k + |B|h(C') + \sum_{t_j \in A} h(t_{j+1} - t_j) \\
&\leq k(2 + h(C')) + |A|h\Big(\frac{\sum_{t_j \in A} t_{j+1} - t_j}{|A|}\Big) \\
&\leq k(2 + h(C')) + |A|h\Big(\frac{p^{n+1}}{|A|}\Big) \\
&= k(2 + h(C')) + C\frac{p^{n+1}}{\log(\frac{p^{n+1}}{|A|})} \\
&\leq k(2 + h(C')) + C\frac{p^{n+1}}{\log(\frac{p^{n+1}}{k})},
\end{aligned}
$$

where the the fourth inequality follows from Jensen's inequality for concave functions. Hence the upper bound is maximized when k is maximized. Thus we want to find an upper bound for the value k. First for each $t < t_1$, we have that

$$\begin{aligned}\sum_{j=1}^{k} b_{j,t} &= e_c(e_{i_{k+1}}, t) - e_c(e_{i_1}, t)\\ &\leq e_c(e_{i_{k+1}}, t_{k+1}) - e_c(e_{i_1}, t)\\ &= e_c(e_{i_{k+1}}, t_{k+1}) - e_c(e_{i_1}, t_1) + e_c(e_{i_1}, t_1) - e_c(e_{i_1}, t)\\ &\leq e_c(e_{i_1}, t_1) - e_c(e_{i_1}, t) + \sum_{j=1}^{k} b_j\\ &= \sum_{j=0}^{k} b_j,\end{aligned}$$

where $b_0 = e_c(e_{i_1}, t_1) - e_c(e_{i_1}, t)$. From the movement bounds we proved earlier and since for each j we assumed that $t_j \geq p^n$, we have that $b_{j,p^{n-1}} > n-3$ for each $j \leq k$. This follows from the fact that the Turing machine head i_j is not erased before p^n steps in the configuration c^{j-1} and p^{n-1} is enough time to have have had any Turing machine head visit $n-3$ cells, even if the simulation area had started from a size 1. Therefore it follows that $\sum_{j=0}^{k} b_j \geq k(n-3)$. The function M^{-1} that we introduced earlier is bounded from below by identity mapping. Therefore we have that

$$\begin{aligned}p^{n+1} &\geq \sum_{j=0}^{k} M^{-1}(b_j)\\ &= k(n-3).\end{aligned}$$

Therefore we have the upper bound $k \leq \frac{p^{n+1}}{n-3}$. Denoting $C'' = \max\{C, h(C')+2\}$ and by combining our inequalities we have that

$$\begin{aligned}\sum_{j=0}^{k} b_j &\leq k(2 + h(C')) + C\frac{p^{n+1}}{\log(\frac{p^{n+1}}{k})}\\ &\leq C''\frac{p^{n+1}}{n-3} + \frac{C''p^{n+1}}{\log(n-3)}\\ &\leq 2C''\frac{p^{n+1}}{\log(n-3)}.\end{aligned}$$

Recall that t_j were assumed to be inside the interval $[p^n, p^{n+1}]$, where $n \geq 5$. For the times less than p^5, we get some constant upper bound C''' and hence taking union of intervals $[p^i, p^{i+1}]$, where $1 \leq i \leq n$, we get our upper bound

$$a(n) = C''' + \sum_{i=5}^{n} C''\frac{p^{i+1}}{\log(i)} \leq C''' + C''\frac{p^{n+2}}{\log(n+1)}.$$

Hence if we choose $m_n = 5M(p^{n+1}) + C''' + C''\frac{p^{n+2}}{\log(n+1)}$, we have that $e_c(i_e, j) = e_{c'}(i_e, j)$ for each $j \leq p^{n+1}$.

Combining all the three cases, we have shown that

$$\frac{I^+_{p^n}(c)}{p^n} \leq \frac{5M(p^n) + 8 + C''' + C''\frac{p^{n+1}}{\log(n)}}{p^n} = \frac{5}{\log(p^n)} + \frac{C''' + 8}{p^n} + \frac{C''p}{\log(n)},$$

which goes to 0 as n goes to infinity. As this holds for each $c \in \Sigma^{\mathbb{Z}}$ and $n \in \mathbb{N}$. Hence we have that $\lambda^{+}(c) = 0$ for each $c \in \Sigma^{\mathbb{Z}}$.

The fact that $\lambda^{-}(c) = 0$ holds for each $c \in \Sigma^{\mathbb{Z}}$ is much easier to see. First of all we have seen that the eraser states travel only to the right direction. Hence if there exists such $i \in \mathbb{N}$ that $c_i = >$, for some $i \geq 0$, it means that if $c' \in W_i^{-}(c)$, then $f^n(c') \in W_i^{-}(f^n(c))$, for each $n \in \mathbb{N}$. Hence any difference coming from right must propagate within a single simulation area. But then it follows from the movement bound that $I_n^{-}(c) \leq M(n)$ and hence $\lambda^{-}(c) = 0$ for each $c \in \Sigma^{\mathbb{Z}}$.

$\square$

Lemma 2. *The cellular automaton (Σ, N, f) constructed in Theorem 2 is sensitive.*

Proof. Let $c \in \Sigma^{\mathbb{Z}}$ and suppose that $I = \{i \in \mathbb{N} \mid f^i(c)_0 = >\}$ is finite. Then there exists such $m \in \mathbb{N}$, that $f^n(c)_0 \neq >$ for each $n \geq m$. Let $k \leq -M(m) - 1$ and $c' \in W_k^{+}(c)$, such that $c'_{[k-1,k]} \in Q_2^1 >$ and $c'_i = a_0$ for each $i < k-1$. We saw in the proof of Lemma 1, that $e_{c'}(k, p^{n+2}) - e_{c'}(k, 0) \geq n$ for each $n \in \mathbb{N}$ and hence there exists such $n \geq m$, that $e_{c'}(k, n) = 0$.

Let us then suppose that $I = \{i \in \mathbb{N} \mid f^i(c)_0 = >\}$ is infinite. Then for each $k < 0$, we choose $c' \in W_k^{+}(c)$, such that $c'_{[k-1,k]} \in Q_2^1 >$ and $c'_i = a_0$ for each $i < k-1$. From the aperiodicity it again follows that there exists such $n \in \mathbb{N}$ that $> \not\subset f^{n'}(c')_{(-\infty,0]}$ for each $n' > n$.

$\square$

Acknowledgements

The author acknowledges the emmy.network foundation under the aegis of the Fondation de Luxembourg for its financial support.

References

1. Blondel, V.D., Cassaigne, J., Nichitiu, C.M.: On the presence of periodic configurations in turing machines and in counter machines. Theor. Comput. Sci. **289**, 573–590 (2002)
2. Bressaud, X., Tisseur, P.: On a zero speed sensitive cellular automaton. Nonlinearity **20**(1), 1–19 (dec 2006). https://doi.org/10.1088/0951-7715/20/1/002
3. Cassaigne, J., Ollinger, N., Torres, R.: A small minimal aperiodic reversible turing machine. Journal of Computer and System Sciences **84** (04 2014). https://doi.org/10.1016/j.jcss.2016.10.004
4. D'amico, M., Manzini, G., Margara, L.: On computing the entropy of cellular automata. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) Automata, Languages and Programming. pp. 470–481. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)
5. Finelli, M., Manzini, G., Margara, L.: Lyapunov exponents vs expansivity and sensitivity in cellular automata. In: ACRI '96. pp. 57–71. Springer London, London (1997)

6. Greiner, W.: Lyapunov Exponents and Chaos, pp. 503–516. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
7. Guillon, P., Salo, V.: Distortion in one-head machines and cellular automata. In: Dennunzio, A., Formenti, E., Manzoni, L., Porreca, A.E. (eds.) Cellular Automata and Discrete Complex Systems. pp. 120–138. Springer International Publishing, Cham (2017)
8. Jeandel, E.: Computability of the entropy of one-tape turing machines. Leibniz International Proceedings in Informatics, LIPIcs **25** (02 2013). https://doi.org/10.4230/LIPIcs.STACS.2014.421
9. Kůrka, P.: Topological Dynamics of Cellular Automata, pp. 9246–9268. Springer New York, New York, NY (2009)
10. Kůrka, P.: On topological dynamics of turing machines. Theoretical Computer Science **174**(1), 203 – 216 (1997). https://doi.org/https://doi.org/10.1016/S0304-3975(96)00025-4, http://www.sciencedirect.com/science/article/pii/S0304397596000254
11. Lyapunov, A.: General Problem of the Stability Of Motion. Control Theory and Applications Series, Taylor & Francis (1992), https://books.google.fi/books?id=4tmAvU3_SCoC
12. Shereshevsky, M.A.: Lyapunov exponents for one-dimensional cellular automata. Journal of Nonlinear Science **2**(1), 1–8 (1992). https://doi.org/10.1007/BF02429850, https://doi.org/10.1007/BF02429850
13. Tisseur, P.: Cellular automata and lyapunov exponents. Nonlinearity **13**(5), 1547–1560 (jul 2000). https://doi.org/10.1088/0951-7715/13/5/308
14. Wolfram, S.: Universality and complexity in cellular automata. Physica D: Nonlinear Phenomena **10**(1), 1 – 35 (1984). https://doi.org/https://doi.org/10.1016/0167-2789(84)90245-8, http://www.sciencedirect.com/science/article/pii/0167278984902458
15. Wolfram, S.: Twenty problems in the theory of cellular automata. Physica Scripta **T9**, 170–183 (jan 1985). https://doi.org/10.1088/0031-8949/1985/t9/029