Preservation of Normality by Non-Oblivious Group Selection

Olivier Carton¹ · Joseph Vandehey²

Published online: 14 August 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract
We give two different proofs of the fact that non-oblivious selection via regular group sets preserves normality. Non-oblivious here means that whether or not a symbol is selected can depend on the symbol itself. One proof relies on the incompressibility of normal sequences, the other on the use of augmented dynamical systems.

Keywords Normal sequences · Selection rules · Automata · Transducers

1 Introduction

An infinite sequence \( x = x_1 x_2 x_3 x_4 \ldots \) over a finite alphabet \( A \) is said to be normal if every finite word appears with the same limiting frequency in \( x \) as every other finite word of the same length. (Fuller definitions of terminology used in the introduction will be included in the next section.)

D.D. Wall [20] famously showed that if \( x = x_1 x_2 x_3 \ldots \) is normal, then \( x_k x_{k+\ell} x_{k+2\ell} x_{k+3\ell} \ldots \) is normal for any \( k, \ell \in \mathbb{N} \). In other words, selecting along an arithmetic progression preserves normality. Furstenberg [8], as part of his seminal paper on the theory of disjoint systems, gave another proof of this fact. Kamae [10] and Weiss [21], using the theory of disjoint systems as well, were able to characterize those sequences of positive integers \( i_1 < i_2 < i_3 < \ldots \) such that selection along these sequences also preserves normality: in particular, these are the deterministic sequences or, equivalently, sequences of Kamae entropy zero.

1 IRIF and Université de Paris, Paris, France
2 The University of Texas at Tyler, Tyler, TX, USA
More generally, many mathematicians have studied (prefix) selection rules. Let $A^*$ be the set of finite words over $A$ and let $L \subset A^*$. The sequence obtained by oblivious selection of $x$ by $L$ is $x \upharpoonright L = x_{i_1}x_{i_2}x_{i_3} \cdots$, where $i_1, i_2, i_3, \ldots$ is the enumeration in increasing order of all the integers $i$ such that the prefix $x_1x_2 \cdots x_{i-1}$ belongs to $L$. This selection rule is called oblivious because the symbol $x_i$ is not included in the considered prefix. If $L = A^*1$ is the set of words ending with a 1, the sequence $x \upharpoonright L$ is made of all symbols of $x$ occurring after a 1 in the same order as they occur in $x$.

The examples above by Wall, Kamae, and Weiss are all examples of oblivious selection rules, where $L$ consists of all words of certain fixed lengths. However, far more intricate selection rules are possible. The following theorem of Agafonov [1] states that normality is preserved by oblivious selection of a regular language.

**Theorem 1** (Agafonov) *If the sequence $x \in A^\infty$ is normal and $L \subset A^*$ is regular, then $x \upharpoonright L$ is also normal.*

A language $L \subset A^*$ is regular if it is accepted by a deterministic finite automaton. We will speak more on this later.

Kamae and Weiss [11] extended Theorem 1 slightly. Let $L$ be a set of words and let $\sim_L$ be an equivalence relation given by $u \sim_L v$ if $\{w : uw \in L\} = \{w : vw \in L\}$. If $L/\sim_L$ is finite, then selection along $L$ preserves normality. In contrast, Merkle and Reimann [12] showed that selection by deterministic one-counter languages or by linear languages need not preserve normality. We also mention that suffix selection, where the selection of a given digit is based on the tail of the sequence after that digit, has also been considered [3].

We can also define the sequence obtained by non-oblivious selection of $x$ by $L$. This is $x \upharpoonright L = x_{i_1}x_{i_2}x_{i_3} \cdots$, where $i_1, i_2, i_3, \ldots$ is the enumeration in increasing order of all the integers $i$ such that the prefix $x_1x_2x_3 \cdots x_i$ including $x_i$ belongs to $L$.

Non-oblivious selection is more powerful than oblivious selection, as it can simulate the latter due to the following formula:

$$x \upharpoonright LA = x \upharpoonright L$$

for any sequence $x$ and any set $L$ of words. Let us recall that $LA$ is the set of words of the form $wa$ for $w \in L$ and $a \in A$. On the other hand, there are weaknesses in non-oblivious selection as well. If we take $L = A^*1$ again, then $x \upharpoonright L$ will consist of nothing but 1’s, which will not be normal.

Accordingly, oblivious selection has been studied more than non-oblivious selection. The second author [19] has a few (very specific) examples of non-oblivious selections that preserve normality. In this paper, we present a more general theorem:

**Theorem 2** *If the sequence $x \in A^\infty$ is normal and $L \subset A^*$ is a regular group set, then $x \upharpoonright L$ is also normal.*

Regular here has the same meaning as in Agafonov’s theorem. Saying that $L$ is a group set implies that in the associated deterministic finite automaton, any input will permute the states.

We note that if there exists a regular group set $K$ such that the symmetric difference $L \Delta K$ is finite, then the non-oblivious selection by $L$ also preserves normality.
More generally, if $L$ is accepted by an automaton such that each recurrent strongly connected component (those components, which, once entered, cannot be left) is a group automaton, then non-oblivious selection by $L$ still preserves normality.

We will prove this result using two distinct methods reflecting the different styles of the two authors of this paper.

The first method, favored by the first author, makes use of the fact that normality can be defined in terms of incompressibility by deterministic finite automata [3]. It follows along the lines of the proof of Agafonov’s theorem presented in [2]. One key ingredient of this proof is the statement that the function which maps each sequence $x$ to the pair $(x \upharpoonright L, x \upharpoonright (A^* \setminus L))$ is one-to-one. The same statement for the non-oblivious selection does not hold, even when $L$ is a group set as is shown by the following example. Consider the set $L$ of words accepted by the automaton pictured in Fig. 1. Let $x$ and $x'$ be the sequences $01^N$ and $101^N$. It is easily computed that $x \upharpoonright L = x' \upharpoonright L = 01^N$ and $x \upharpoonright (A^* \setminus L) = x' \upharpoonright (A^* \setminus L) = 1^N$.

The second method, favored by the second author, makes use of the idea of augmented systems, dynamical systems which have been extended to simultaneously act over a deterministic finite automaton [9, 18]. We make use of a technique from [19], where we use an automaton that also records a finite number of selected symbols: this reduces the problem of counting frequencies of words in $x \upharpoonright L$ to the problem of calculating visiting frequencies of certain states in the automaton.

In Section 2, we will give definitions, notation, and the necessary results from previous papers. In Section 3, we provide a proof of Theorem 2, following the ideas of incompressibility. In Section 4, we provide a proof of Theorem 2, following the ideas of augmented systems.

2 Preliminaries

2.1 Sequences, Words, and Normality

We write $\mathbb{N} = \{1, 2, 3, \ldots\}$ for the set of all natural numbers. An alphabet $A$ is a finite set with at least two symbols. We respectively write $A^*$ and $A^\mathbb{N}$ for the set of all finite sequences (also known as words) and the set of all infinite sequences of elements of $A$ (which we will simply refer to as sequences). We also write $A^k$ stands for the set of all words of length $k$. The length of a finite word $w$ is denoted by $|w|$. The empty word is denoted by $\lambda$. The positions in finite and infinite sequences are numbered starting from 1. For a word $w$ and positions $1 \leq i \leq j \leq |w|$, we let $w[i]$ and $w[i..j]$ denote the symbol $a_i$ at position $i$ and the word $a_ia_{i+1}\cdots a_j$ from position $i$ to position $j$. A set $C$ of words is called prefix-free if for any $u, v \in C$ with $u$ being a prefix of $v$ ($u = v[1..|u|]$), we have $u = v$. We write $\log$ for the base 2 logarithm. For any finite set $S$ we denote its cardinality with $\#S$.

Fig. 1 A group automaton with a final state at $q_1$
Let \( x = a_1a_2a_3 \cdots \) be a sequence over the alphabet \( A \). Let \( L \subseteq A^* \) be a set of words over \( A \). As described in the introduction, the sequence obtained by oblivious selection of \( x \) by \( L \) is \( x \upharpoonright L = a_{i_1}a_{i_2}a_{i_3} \cdots \), where \( i_1, i_2, i_3, \ldots \) is the enumeration in increasing order of all the integers \( i \) such that the prefix \( a_1a_2 \cdots a_{i-1} \) belongs to \( L \). The sequence obtained by non-oblivious selection of \( x \) by \( L \) is \( x \upharpoonright \upharpoonright L = a_{i_1}a_{i_2}a_{i_3} \cdots \) where \( i_1, i_2, i_3, \ldots \) is the enumeration in increasing order of all the integers \( i \) such that the prefix \( a_1a_2a_3 \cdots a_i \) including \( a_i \) belongs to \( L \).

We recall here the notion of normality. We start with the notation for the number of occurrences of a given word \( u \) within another word \( w \). For \( w \) and \( u \) two words, the number \( |w|_u \) of occurrences of \( u \) in \( w \) is given by \( |w|_u = \#\{i : w[i..i+|u|-1] = u\} \).

We say that \( x \in A^\mathbb{N} \) is normal if for each word \( w \in A^* \), we have
\[
\lim_{n \to \infty} \frac{|x[1..n]|_w}{n} = (\#A)^{-|w|}.
\]

This differs from Borel’s original definition [5] of normality, but is equivalent (see [2, Sect. 7.3]).

### 2.2 Deterministic Finite Automata

A deterministic finite automaton is a tuple \( T = \langle Q, A, \delta, I, F \rangle \), where \( Q \) is a finite set of states, \( A \) is the input alphabet, \( \delta : Q \times A \to Q \) is the transition function, and \( I \subseteq Q \) and \( F \subseteq Q \) are the sets of initial and final states, respectively. We focus on automata that operate in real-time, that is, they process exactly one input alphabet symbol per transition. Moreover, we will assume that there is a single initial state, that is, \( I \) is a singleton set.

The relation \( \delta(p, a) = q \) is written \( p \xrightarrow{a} q \) and we further denote the sequence of consecutive transitions
\[
q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \cdots \xrightarrow{a_{n-1}} q_n
\]
by \( q_0 \xrightarrow{a_1a_2 \cdots a_n} q_n \). A word \( w = a_1a_2 \cdots a_n \) is said to be accepted by an automaton if \( q_0 \xrightarrow{w} q_n \), \( q_0 \) is the initial state (that is, \( I = \{q_0\} \)), and \( q_n \) is final (that is, in \( F \)). A language \( L \subseteq A^* \) is said to be regular (as seen in Theorem 1) if there exists a deterministic finite automaton \( T \) such that \( w \in L \) if and only if \( w \) is accepted by \( T \).

We now introduce a classical class of regular sets called group sets (as seen in Theorem 2). A group automaton is a deterministic automaton such that each symbol induces a permutation of the states. By inducing a permutation, we mean that, for each symbol \( a \), the function which maps each state \( p \) to the state \( q \) such that \( p \xrightarrow{a} q \) is a permutation of the state set. Put another way, if \( p \xrightarrow{a} q \) and \( p' \xrightarrow{a} q \) are two transitions of the automaton, then \( p = p' \). A regular set \( L \subseteq A^* \) is called a group set if \( L \) is accepted by a group automaton. It is well known that a regular set is a group set if and only if its syntactic monoid is a group [15, Sect. 7.5].
2.3 Dynamical Systems

We will consider dynamical systems to consist of a tuple $\mathcal{X} = \langle X, \mathcal{F}, T, \mu \rangle$, where $X$ is a space, $\mathcal{F}$ is a $\sigma$-algebra on this space, $T : X \to X$ is a transformation (always assumed to be measurable with respect to $\mathcal{F}$), and $\mu$ is a measure on $\mathcal{F}$. We say that $T$ preserves the measure $\mu$, or, equivalently, that $\mu$ is $T$-invariant, if $\mu(T^{-1}A) = \mu(A)$ for all $A \in \mathcal{F}$. We say that a system is ergodic if $\mu(T^{-1}A \Delta A) = 0$ for some $A \in \mathcal{F}$ implies that $\mu(A) = 0$ or $\mu(X \setminus A) = 0$. Ergodicity may be considered an indecomposability criterion for dynamical systems: a system is ergodic if it cannot be split into two large $T$-invariant pieces.

Of particular relevance to this paper is the symbolic shift system. Let $X = A^\mathbb{N}$ be the space of all infinite words on the alphabet $A$. Let $T$ denote the forward shift on $X$, so that for a sequence $x \in X$, $(Tx)_i = x_{i+1}$. Given a word $w \in A^*$, let $C_w$ denote the cylinder set corresponding to $w$, so that $C_w$ consists of all $x \in X$ such that $x[1..|w|] = w$. The cylinder sets form a semi-algebra that generates the canonical $\sigma$-algebra on $X$ and so if we let $\mu$ be a measure on cylinder sets $C_w$ given by $\mu(C_w) = (#A)^{-|w|}$, then this extends to a measure on this $\sigma$-algebra. We note that if $A = \{0, 1, \ldots, b - 1\}$ and we use the standard bijection\(^1\) from $X$ to $[0, 1)$ associating an infinite word with a $b$-ary expansion, then $\mu$ is just the Lebesgue measure.

We may reinterpret the definition of normality in a slightly more ergodic manner. Since $x[\ldots i + |w| - 1] = w$ if and only if $T^{i-1}x \in C_w$, we have that a word $x \in X$ is normal if for each finite word $w \in A^*$, we have

$$\lim_{n \to \infty} \frac{\# \{0 \leq i \leq n-1 : T^i x \in C_w\}}{n} = \mu(C_w).$$

Note that $T$ is ergodic and invariant with respect to $\mu$, so by the pointwise ergodic theorem, $\mu$-almost all $x \in X$ are normal.

By the Pyatetskii-Shapiro normality criterion [13], we can weaken the above to say that $x$ is normal if there exists a fixed constant $c > 0$ such that for every $w \in A^*$, we have

$$\limsup_{n \to \infty} \frac{\# \{0 \leq i \leq n-1 : T^i x \in C_w\}}{n} \leq c \mu(C_w).$$

3 The First Method of Proof

We now introduce automata with output, also known as transducers, which are used to compress sequences and to select symbols from a sequence. In this paper we only consider input-deterministic transducers (also known as sequential) computing functions from sequences to sequences. Such a machine is a deterministic automaton in which each transition is equipped with an additional output word. The output of a (infinite) run is the concatenation of the outputs of the transitions used by the run. More formally a transducer is a tuple $\mathcal{T} = \langle Q, A, B, \delta, I, F \rangle$, where $Q^\text{\footnote{Actually not quite a bijection, due to the phenomenon of 0.09 = 0.1, but this will not be relevant and can be ignored.}}$
is a finite set of states, $A$ and $B$ are the input and output alphabets, respectively, $\delta : Q \times A \to B^* \times Q$ is the transition function, and $I \subseteq Q$ and $F \subseteq Q$ are the sets of initial and final states, respectively. The set $I$ is again a singleton set.

The relation $\delta(p, a) = (w, q)$ is written $p \xrightarrow{a,w} q$ and the tuple $\langle p, a, w, q \rangle$ is then called a transition of the transducer. A finite (respectively, infinite) run is a finite (respectively, infinite) sequence of consecutive transitions,

$$q_0 \xrightarrow{a_1|v_1} q_1 \xrightarrow{a_2|v_2} \cdots \xrightarrow{a_n|v_n} q_n.$$

Its input and output labels are respectively $a_1 \cdots a_n$ and $v_1 \cdots v_n$. A finite run is written $q_0 \xrightarrow{a_1 \cdots a_n|v_1 \cdots v_n} q_n$. An infinite run is final if the state $q_n$ is final for infinitely many integers $n$. In that case, the infinite run is written $q_0 \xrightarrow{a_1 a_2 a_3 \cdots |v_1 v_2 v_3 \cdots} \infty$. An infinite run is accepting if it is final and furthermore its first state $q_0$ is the initial one. This is the classical Büchi acceptance condition [14]. Since transducers are supposed to be input-deterministic, there is at most one accepting run $q_0 \xrightarrow{x|y} \infty$ having a given sequence $x$ for input label and we write $y = T(x)$.

A transducer is called one-to-one if the function which maps $x$ to $y$ is one-to-one. We always assume that all transducers are trim: each state occurs in at least one accepting run.

A sequence $x = a_1 a_2 a_3 \cdots$ is compressible by a transducer $T$ if it has an accepting run $q_0 \xrightarrow{a_1|v_1} q_1 \xrightarrow{a_2|v_2} q_2 \xrightarrow{a_3|v_3} q_3 \cdots$ satisfying

$$\liminf_{n \to \infty} \frac{|v_1 v_2 \cdots v_n| \log \#B}{n \log \#A} < 1.$$ 

Recall that each of the $v_i$’s belongs to $B^*$, not necessarily $B$, so could be empty or have length greater than 1.

The connection between compressible sequences and normality is given by the following:

**Theorem 3** A sequence is normal if and only if it not compressible by a one-to-one deterministic transducer.

The above result follows from the results in [7, 16]. A direct proof appears in [4]. Extensions of this characterization for non-deterministic and extra memory transducers are in [3, 6].

Let $c : A^k \to A^*$ be a function mapping each word of length $k$ to some word. This function can be extended to a function from $(A^k)^*$ to $A^*$ by setting $c(w_1 \cdots w_n) = c(w_1) \cdots c(w_n)$ with $w_i \in A^k$ for $1 \leq i \leq n$. When a sequence $x$ is not normal, it can be compressed using a Huffman coding. This is implicit in the following lemma. The proof of the next lemma is the first part of the proof of Lemma 7.5.1 in [2].
Lemma 4 If the sequence \( x \) is not normal, there is a length \( k \) and a one-to-one function \( c : A^k \to C \) where \( C \) is a prefix-free set such that
\[
\lim_{n \to \infty} \frac{|c(x[1..nk])|}{nk} < 1.
\]

The next lemma states that if the input \( x \) is normal, each state which is visited infinitely often in the run over \( x \) in some deterministic automaton is visited at least a linear number of times. This lemma is actually Lemma 7.10.3 in [2].

Lemma 5 Let \( x = a_0a_1a_2 \cdots \) be a normal sequence and let \( q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} \cdots \) be a run in a deterministic automaton. If the state \( q \) is visited infinitely often in this run, then
\[
\lim_{n \to \infty} \frac{\#\{i \leq n : q_i = q\}}{n} > 0.
\]

A transducer can have two output tapes. In this case, the transducer would be a tuple \( T = \langle Q, A, B, \delta, I, F \rangle \), where now the transition function is \( \delta : Q \times A \to B^* \times B^* \times Q \). The relation \( \delta(p, a) = (w_1, w_2, q) \) is written \( p \xrightarrow{a|w_1,w_2} q \).

A deterministic automaton \( A \) accepting a set \( L \) can be turned into a two-output transducer \( T \) that outputs \( x \restriction L \) and \( x \restriction (A^* \setminus L) \) on its first and second output tapes respectively. Each transition \( p \xrightarrow{a} q \) of \( A \) is replaced by either the transition \( p \xrightarrow{a|a,\lambda} q \) if the state \( q \) is final or by the transition \( p \xrightarrow{a|\lambda,a} q \) if \( q \) is not final. If \( q \) is final, any finite run in \( A \) from the initial state to \( q \) is accepting and therefore the label \( a \) of its last transition must be output to the first output tape because it is selected in \( x \restriction L \).

The following lemma states the key property of this transducer \( T \) when \( A \) is a group automaton.

Lemma 6 Let \( A \) be a group automaton and let \( T \) the transducer obtained from \( A \) as above. The function which maps each finite run \( p \xrightarrow{u,v,w} q \) of \( T \) to the triple \( q,v,w \) is one-to-one.

**Proof** Note first that \( |u| = |v| + |w| \). The proof is done by induction on the sum \( |v| + |w| \). We will, throughout the proof, denote the run \( p \xrightarrow{u,v,w} q \) by \( \rho \).

If \( v \) and \( w \) are empty, \( u \) is also empty and so \( \rho \) must be the empty run from \( p = q \) to \( p \). Suppose now that \( |v| + |w| > 0 \). Suppose that \( q \) is final in the automaton \( A \). All transitions ending in \( q \) have the form \( r \xrightarrow{a|a,\lambda} q \) for some state \( r \) and some symbol \( a \). It follows that \( v \) cannot be empty. Let \( a \) be the last symbol of \( v \) and let \( v' \) be such that \( v = v'a \). Since \( A \) is a group automaton, there is exactly one transition \( r \xrightarrow{a} q \) ending in \( q \) and having \( a \) for label. This implies that the last transition of \( \rho \) is \( r \xrightarrow{a|a,\lambda} q \). Applying the induction hypothesis to the triple \( \langle r, v', w \rangle \) completes the proof in this case. The case where \( q \) is not final in \( A \) works similarly.

**Proof of Theorem 2** Let \( x \) be a normal word. Let \( L \subset A^* \) be a regular group set. We suppose that \( x \restriction L \) is not normal and will show that \( x \) can be compressed by a one-to-one deterministic transducer, contradicting its normality.
First note that $x \upharpoonright (A^* \setminus L)$ cannot be a finite word, as otherwise the normality of $x \upharpoonright L$ is trivial.

Let $A$ be a group automaton accepting $L$ whose state set and transition function are $Q$ and $\delta$ respectively. Let $T_1$ be the two-output transducer obtained from $A$ as above. The state set of $T_1$ is the same as the one of $A$ and the transition function $\delta_1$ of $T_1$ is defined as follows.

$$\delta_1(p, a) = \begin{cases} (a, \lambda, \delta(p, a)) & \text{if } \delta(p, a) \in F \\ (\lambda, a, \delta(p, a)) & \text{if } \delta(p, a) \notin F. \end{cases}$$

Since this transducer is assumed to be trim and since $x$ is assumed to be normal, it must reach final states infinitely often, showing that $x \upharpoonright L$ is infinite. This is a consequence of [16, Satz 2.5].

Since it is supposed that $y = x \upharpoonright L$ is not normal, there is, by Lemma 4, an integer $k$ and a one-to-one function $c$ from $A^k$ into a prefix-free set $C$ such that

$$\liminf_{n \to \infty} \frac{|c(y[1..nk])|}{nk} < 1.$$
comes from $B_1$ and 1 otherwise. To be more precise, the state set of $T_3$ is $Q \times B(k) \times B(m) \times B(m)$ and its transition function $\delta_3$ is defined as follows.

$$
\delta_3((p, w_1, w_2, w_3), a) =
\begin{cases}
(\lambda, (\delta(p, a), w_1a, w_2, w_3)) & \text{if } \delta(p, a) \in F, |w_1a| < k \\
(\lambda, (\delta(p, a), \lambda, w_2c(w_1a), w_3)) & \text{if } \delta(p, a) \in F, |w_1a| = k \\
(0u, (\delta(p, a), \lambda, v, w_3)) & \text{if } \delta(p, a) \notin F, |w_3a| < m \\
\end{cases}
$$

We assume that $m$ is sufficiently large so that $|c(w)| \leq m$ for any $w \in A^k$: this guarantees that in the third case above, we have $m \leq w_2c(w_1a) < 2m$ so $v$ is in $B(m)$ as desired. Moreover, we claim that, for $m$ great enough, the transducer $T_3$ also compresses its input $x$. Note that the output of $T_3$ is longer than the sum of the two outputs of $T_2$ because each block of $m$ symbols is preceded by an extra bit 0 or 1. However, for $m$ great enough, this loss is offset by the compression of $T_2$. Specifically, suppose we let $m_0 = \max\{k, \max_{w \in A^k} |c(w)|\}$ and let $u_n$ denote the output of $T_3$ after $n$ transitions. Let $u_n, v_n$ be the outputs of the two tapes of $T_2$ after $n$ transition, as in the previous paragraph. Then we have that

$$
|u_n| - (|u_n| + |v_n|) \leq \frac{m_0n}{nm}.
$$

This inequality comes from the fact that the length of the outputs of $T_2$ are increased by a maximum of $m_0$ every time we increase $n$ by $k$, and the extra bit added by $T_3$ occurs at most once every time the outputs of $T_2$ are increased by $m$. The inequality then implies that

$$
\liminf_{n \to \infty} \frac{|u_n|}{n} \leq \liminf_{n \to \infty} \frac{|u_n| + |v_n|}{n} + O\left(\frac{1}{m}\right)
$$

and so by choosing $m$ sufficiently large, we see that $T_3$ compresses its input.

Note that none of the transducers $T_1, T_2$, and $T_3$ is one-to-one because the function which maps $x$ to the pair $(x \parallel L, x \parallel (A^* \setminus L))$ might not be one-to-one. For that reason, we construct a last transducer $T_4$ obtained by changing $T_3$ to make it one-to-one. The transducer $T_4$ works as $T_3$ but whenever $T_3$ writes to its output tape a block of length $m$ coming from its buffer $B_1$ with its extra bit 0, the transducer $T_4$ also writes some extra information that we now describe. This extra information is made of two data. The first one is the current state $\delta(p, a)$ of the automaton $A$. This other one is the length of the buffer $B_2$. Both data are written in binary and require $\lceil \log \#Q \rceil$ and
[\log m] \text{ bits respectively. We do not give explicitly the transition function of } \mathcal{T}_4 \text{ as it is almost the same as the one of } \mathcal{T}_3.

If we let \( u_n, v_n, \tilde{u}_n \) have the same meaning as before and let \( \tilde{u}_n \) denote the output of \( \mathcal{T}_4 \) after \( n \) transitions. Then we have that

\[
|\tilde{u}_n| - (|u_n| - |v_n|) \leq \frac{m_0 n}{km} \left( 1 + \lceil \log \#Q \rceil + \lceil \log m \rceil \right).
\]

Thus, we have that

\[
\liminf_{n \to \infty} \frac{|\tilde{u}_n|}{n} \leq \liminf_{n \to \infty} \frac{|u_n| + |v_n|}{n} + O \left( \frac{\log m}{m} \right),
\]

and so \( \mathcal{T}_4 \) still compresses its input for \( m \) large enough. We also claim that the transducer \( \mathcal{T}_4 \) is one-to-one. To do this, we show that, from the output of \( \mathcal{T}_4 \), it is possible to recover the input.

Consider any block in the output which comes from \( B_1 \) (indicated by the first binary bit in front of it). If we take this block and all preceding blocks that arise from \( B_1 \), apply \( c^{-1} \) to them, and concatenate them in order, we obtain a prefix \( v \) of \( x \parallel L \). In a similar way, if we take all preceding blocks that arise from \( B_2 \) and the numbers of symbols of the next \( B_2 \) block indicated by the second data, we obtain a prefix \( w \) of \( x \parallel (A^* \setminus L) \). (Recall our assumption that \( x \parallel (A^* \setminus L) \) is infinite, so this next block must always exist.) Finally, the first data gives us a state \( q \) for our original automaton \( A \). From the way that \( \mathcal{T}_3 \) is constructed, this specific triple \( \langle q, v, w \rangle \), represents the outputs and state reached by a run \( q_0 \xrightarrow{u|v,w} q \). By Lemma 6, the triple \( \langle q, v, w \rangle \) uniquely identifies this \( u \) and means it must necessarily be a prefix of \( x \). Since there are infinitely many blocks that come from \( B_1 \), this gives us infinitely many prefixes of \( x \) and so we know the entirety of \( x \).

4 The Second Method of Proof

We now consider what happens when we augment a dynamical system to simultaneously run over a finite state automaton. Let us consider, as before, a dynamical system \( \langle X, \mathcal{F}, T, \mu \rangle \) that is a symbolic shift and an automaton \( \langle Q, A, \delta, I, F \rangle \). We will say this automaton is transitive if given any \( q_1, q_2 \in Q \) there exists a word \( w \in A^* \) such that \( q_1 \xrightarrow{w} q_2 \).

We now consider the following augmented dynamical system \( \langle \tilde{X}, \tilde{\mathcal{F}}, \tilde{T}, \tilde{\mu} \rangle \):

1. \( \tilde{X} := X \times Q \),
2. \( \tilde{T} : \tilde{X} \to \tilde{X} \) given by \( \tilde{T}(x, q) = (T x, \delta(q, x_1)) \), where \( x = x_1 x_2 x_3 \ldots \),
3. Cylinder sets \( C_{w,q} = C_w \times \{ q \} \) for \( w \in A^*, q \in Q \) (noting again that these cylinder sets generate a \( \sigma \)-algebra \( \tilde{\mathcal{F}} \) on \( \tilde{X} \)),
4. \( \tilde{\mu}(C_{w,q}) = (\#A)^{-|w|}/(\#Q) \).

We extend our definition of normality on augmented systems: \( (x, p) \in \tilde{X} \) is said to be normal if for every \( w \in A^* \) and \( q \in Q \), we have

\[
\lim_{n \to \infty} \frac{\# \{ 0 \leq i \leq n - 1 : T^i(x, p) \in C_{w,q} \}}{n} = \tilde{\mu}(C_{w,q}).
\]
We have then the following result:

**Theorem 7** Suppose that the automaton \( \langle Q, A, \delta, F \rangle \) is a transitive automaton. If \( \tilde{T} \) preserves the measure \( \tilde{\mu} \), then \( \tilde{T} \) is ergodic. Moreover, for any \( x \in X \) that is normal, the point \( (x, q) \) is normal w.r.t \( \tilde{\mu} \) for any \( q \in Q \).

This is a simplified version of Theorem 3.1 in [18] (see Remark 3.2 in that paper for the discussion of the necessary conditions needed on the dynamic system and note that they are all trivial in our case). See also [17] for a simpler proof.

We again want to consider adding a buffer to an automaton; however, unlike in the previous section, we wish to consider a “rolling” buffer, which will continuously record the previous few inputs that caused us to reach a final state without resetting itself to a shorter word.

For a given \( k \in \mathbb{N} \) and automaton \( \langle Q, A, \delta, I, F \rangle \) as above, consider the automaton \( \langle Q_k, A, \delta_k, I_k, F_k \rangle \), where \( \delta_k \) satisfies the following rules:

- If \( \delta(q, a) \) is not final, then \( \delta_k((q, w), a) = (\delta(q, a), w) \).
- If \( \delta(q, a) \) is final, then \( \delta_k((q, w), a) = (\delta(q, a), w[2..k-1]a) \),

and \( I_k = \{(q_0, w_0)\} \), where \( w_0 \) is any element of \( A^k \). The choice of which \( w_0 \) to use will not be relevant for any subsequent proofs. We will refer this new automaton as a \( k \)-digit buffer over the original automaton.

**Lemma 8** Let \( x \in X \) be normal and \( q \in Q_k \), and let \( \langle Q_k, A, \delta_k, I_k, F_k \rangle \) be a \( k \)-digit buffer over a transitive group automaton. Then there is a subset \( Q' \subseteq Q_k \) such that \( F_k \cap Q' \) is non-empty, \( \langle Q', A, \delta_k, I_k \cap Q', F_k \cap Q' \rangle \) is a transitive automaton, and \( \tilde{T}^i(x, q) \) will eventually always be in \( Q' \) in its second coordinate. Moreover, every word in \( A_k \) will appear in the second coordinate of some element of \( Q' \).

We are being somewhat imprecise about the initial states in the new automaton \( \langle Q', A, \delta_k, I_k \cap Q', F_k \cap Q' \rangle \). It is possible that \( I_k \cap Q' = \emptyset \). In this case we would replace \( I_k \cap Q' \) with the first state in \( Q' \) that the \( \tilde{T} \)-orbit of \( (x, q) \) enters—in essence, shifting everything forward.

**Proof** Here we adapt the method of Lemma 4.1 from [18].

First, note that a \( k \)-digit buffer over a transitive group automaton need not be a transitive automaton (nor even a group automaton) in its own right. So given a subset \( Q'_k \subseteq Q_k \), we say that this subset is a transitive component if \( \langle Q'_k, A, \delta_k \rangle \) is a transitive automaton. In particular, for any \( (q_1, w_1), (q_2, w_2) \in Q'_k \), there exists a word \( u \in A^* \) such that \( (q_1, w_1) \overset{u}{\rightarrow} (q_2, w_2) \), and for any \( (q_1, w_1) \in Q'_k \) and any word \( u \in A^* \), we have that \( (q_1, w_1) \overset{u}{\rightarrow} (q_2, w_2) \) for some \( (q_2, w_2) \in Q'_k \). In particular, these properties imply that if \( Q'_k \) are both transitive components and \( Q'_1 \cap Q'_2 \neq \emptyset \), then \( Q'_1 = Q'_2 \). As such there is exactly one way to partition \( Q_k \) into a finite collection of transitive components \( Q^1_k, Q^2_k, \ldots, Q^m_k \) together with a finite
collection of elements \((q_1, w_1), \ldots, (q_n, w_n)\) that do not belong to any transitive component. We will refer to these latter elements as free elements of \(Q_k\).

For the remainder of this proof, we will say that \((q, w)\) connects to \((q', w')\) if there exists some word \(u \in A^*\) such that \((q, w) \xrightarrow{u} (q', w')\). In this case we would say that \((q, w)\) connects to \((q', w')\) via \(u\).

We now claim that any free element \((q, w)\) of \(Q_k\) connects to an element \((q', w')\) that belongs to a transitive component of \(Q_k\). Suppose, by way of contradiction, that this fails for some free element \((q, w)\). (If there are several such free elements, we will choose one that minimizes the number of other elements it connects to.) Therefore the only elements of \(Q_k\) that \((q, w)\) can connect to are other free elements. If it also connects to itself via a non-empty word, then it is clear that \((q, w)\), together with the set of elements it connects to, form a transitive component, contrary to our assumption of \((q, w)\) being free. On the other hand, if \((q, w)\) does not connect to itself by a non-empty word, let \(a\) be a fixed element of \(A\) and let \((q_a, w_a) = \delta((q, w), a)\).

By our assumption on \((q, w)\), we have that \((q_a, w_a)\) too cannot connect to any element in a transitive component. If \((q_a, w_a)\) connects to itself, then by the same argument as before, \((q_a, w_a)\) lies in a transitive component, which is a contradiction. Otherwise, we now note that the set of elements \((q_a, w_a)\) connects to must be a proper subset of the set of elements that \((q, w)\) connects to, which contradicts our minimality assumption of \((q, w)\). Since we have reached a contradiction in every case, we must have that any free element connects to an element that belongs to a transitive component.

We will now define a sequence of words \(u_i\), for \(1 \leq i \leq n\), inductively. First let \(u_1\) be any word such that \((q_1, w_1)\) connects to an element of a transitive component via \(u_1\). Then, if \(u_1, \ldots, u_{i-1}, i < n\), have been defined, we define \(u_i\) as follows. If \((q_i, w_i)\) connects to an element of a transitive component via the concatenated word \(u_1u_2 \ldots u_{i-1}\), then \(u_i\) is the empty word. Otherwise, \((q_i, w_i)\) connects to some free element \((q'_i, w'_i)\) via \(u_1u_2 \ldots u_{i-1}\), and so we let \(u_{i+1}\) be defined so that \((q'_i, w'_i)\) connects to an element of a transitive component via \(u_{i+1}\). With all terms defined in this way we see that any free element \((q_i, w_i)\) connects to an element of a transitive component via the word \(u_1u_2 \ldots u_i\).

Let \(u = u_1u_2 \ldots u_n\). Since \(x\) is assumed to be normal, \(u\) must appear in the expansion of \(x\) at some point. Let us say the first appearance occurs starting at \(x_L\). It is clear then from the construction of \(u\) that if the second coordinate of \(\widehat{T}^{\ell-1}(x, q)\) does not belong to a transitive component, then there must exist some \(i \leq |u|\) such that the second coordinate of \(\widehat{T}^{\ell-1+i}(x, q)\) does belong to a transitive component. Regardless, this tells us that the second coordinate of the \(\widehat{T}\)-orbit of \((x, q)\) eventually enters some transitive component and remains there. We will refer to this particular transitive component as \(Q'\).

Now consider any \((q_1, w) \in Q'\) and \(q_2 \in Q\). Since \(\langle Q, A, \delta, I, F \rangle\) is transitive, there exists another word, \(u \in A^*\), so that \(q_1 \xrightarrow{u} q_2\) in this automaton. Moreover, since \((Q_k, A, \delta_k, I_k, F_k)\) behaves the same as \((Q, A, \delta, I, F)\) in the first coordinate, we must have that \((q_1, w) \xrightarrow{u} (q_2, w_2)\) for some word \(w_2 \in A^k\). Since \(q_2\) is an arbitrary element of \(Q\), we have that every element of \(Q\) appears in the first coordinate of \(Q'\). In particular, \(F_k \cap Q'\) is non-empty.
We will finally show that every word in $A^k$ will appear in the second coordinate of some element of $Q'$. Note that if one starts at a state $(q, w) \in F_k \cap Q'$, it is always possible to reach the next final state in $F_k \cap Q'$ via any element in $A$. In particular, $(Q, A, \delta, I, F)$ itself is a group automaton and so the action of any input is to permute the states. Thus if one starts at $q$ and keeps repeating the input $a_1$, one must eventually arrive at state $q_1 \in F$. At worst the permutation that $a_1$ induces on the states $Q$ has $q$ being the only final state in its cycle. But even in this case, we would just have $q_1 = q$. Thus, in our buffered automaton, we have that by inputting $a_1$ enough times we will move from $(q, w)$ to $(q_1, w[2..k]a_1) \in Q'$. We may repeat this process by inputting $a_2$ over and over until we reach a state $(q_2, w[3..k]a_1a_2) \in Q'$, and so on, until we reach $(q_k, a_1a_2 \ldots a_k) \in Q'$. But since the $a_i$’s are all arbitrary, one can force any desired word to appear in the second coordinate of $Q'$.

Note that since transitive components are disjoint, $(x, q)$ completely determines $Q'$. Namely, $Q'$ is the transitive component that the $\hat{T}$-orbit of $(x, q)$ eventually enters in the second component.

**Lemma 9** Suppose we augment $(X, T, \mu)$ with $(Q', A, \delta_k, I_k, F_k)$ as in the previous lemma. Then $T$ preserves the measure $\hat{\mu}$.

We will use a similar method to the proofs seen in [19].

**Proof** Any set $E \subset \hat{X}$ can be decomposed as as $E = \bigcup_{(q, w) \in Q'} E_{q, w} \times \{(q, w)\}$. Since the sets $\hat{T}^{-1} \left( E_{q, w} \times \{(q, w)\} \right)$ are all disjoint, if we can show that $\hat{T}$ preserves the $\hat{\mu}$-measure of sets of the form $E_{q, w} \times \{(q, w)\}$ then it will follow that $\hat{T}$ preserves the $\hat{\mu}$-measure of $E$, and we are done.

To prove this, consider the inverse branches $T^{-1}_a$, $a \in A$, of $T$, such that $T^{-1}_a x = ax$. We may then likewise decompose $\hat{T}^{-1}$ into branches $\hat{T}^{-1}_a$, $a \in A$, such that $\hat{T}^{-1}_a$ induces the branch $T^{-1}_a$ in the first coordinate. We may then analyze exactly how $\hat{T}^{-1}_a$ acts: in particular, $\hat{T}^{-1}_a(x, (q, w))$ equals

$$\begin{cases} 
(ax, (\delta^{-1}(q, a), w)), & \text{if } q \notin F, \\
(ax, (\delta^{-1}(q, a), Aw[1..k - 1])), & \text{if } q \in F \text{ and } w[k] = a \\
\emptyset, & \text{if } q \in F \text{ and } w[k] \neq a 
\end{cases}$$

where $\delta^{-1}(q, a)$ is the unique state $p \in Q$ such that $\delta(p, a) = q$. (Unique due to $(Q, A, \delta, F)$ being a group automaton.)

Since $\mu(aE_{q, w}) = \mu(E_{q, w})/|A|$ and since for any subset $Y \subset X$ we have that $\hat{\mu}(Y \times \{(q, w)\}) = \mu(Y)/|Q'|$, we therefore have that $\hat{\mu}(\hat{T}^{-1}_a(E_{q, w} \times \{(q, w)\}))$ equals

$$\begin{cases} 
\frac{\mu(E_{q, w})}{|A|} \times \frac{1}{|Q'|}, & \text{if } q \notin F, \\
\frac{\mu(E_{q, w})}{|A|} \times \frac{|A|}{|Q'|}, & \text{if } q \in F \text{ and } w[k] = a, \\
0, & \text{if } q \in F \text{ and } w[k] \neq a. 
\end{cases}$$
By summing over all the $a$’s, we see that
\[
\tilde{\mu}(\tilde{T}^{-1}(E_{q,w} \times \{(q,w)\})) = \frac{\mu(E_{q,w})}{\#A} \times \frac{\#A}{\#Q'} = \frac{\mu(E_{q,w})}{\#Q'} = \tilde{\mu}(E_{q,w} \times \{(q,w)\})
\]
in all cases, which completes the proof.

Second proof of Theorem 2 Let $x = a_1a_2a_3 \cdots \in X$ be normal and let $y = x \parallel L = b_1b_2b_3 \ldots$. Let $(Q, A, \delta, I, F)$ denote a group automaton that accepts $L$.

Consider any finite word $w \in A^*$ with length $k = |w|$. By the Pyatetskii-Shapiro normality criterion, we want to show that there exists a uniform $c > 0$ (independent of our choice of $w$) such that
\[
\limsup_{m \to \infty} \frac{\#\{0 \leq i \leq m - 1 : T^i y \in C_w\}}{m} \leq c \mu(C_w).
\]

We can analyze how often $w$ appears in $y$ by analyzing the behavior of $x$ when lifted to a augmented system with the $k$-digit buffer $(Q_k, A, \tilde{\delta}, I_k, F_k)$. The particular lift we choose is $\tilde{x} = (x, (q_0, w_0))$. Let $i_1, i_2, i_3, \ldots$ be the increasing sequence of indices $i$ such that $\tilde{T}^i \tilde{x} \in X \times F_k$, starting with the $k$th such index. Then if we let $\pi : \tilde{X} \to A^k$ be the projection onto the length-$k$ word contained in the second coordinate, then we see that $y[j..j+k−1] = \pi(\tilde{T}^i \tilde{x})$.

By Lemma 8, we know there is a subset $Q'_k \subseteq Q_k$ such that eventually the orbit $\tilde{T}^i \tilde{x}$ will always be in $X \times Q'_k$. By, as necessary, ignoring a finite piece of $x$, we may assume that we are always in $X \times Q'_k$. Since by Lemma 9, $\tilde{T}$ preserves the measure $\tilde{\mu}$, it must also preserve the measure of $\tilde{\mu}$ when restricted to $X \times Q'_k$. Therefore by Theorem 7, $\tilde{T}$ restricted to $X \times Q'_k$ is ergodic and invariant with respect to the restriction of $\tilde{\mu}$.

Let $\tilde{C}_w$ denote the subset of $Q'_k \cap F_k$ such that the second coordinate is $w$. By the last part of Lemma 8, this is always non-empty.

Then, with this definition, we have that
\[
\limsup_{m \to \infty} \frac{\#\{0 \leq i \leq m - 1 : \tilde{T}^i \tilde{x} \in X \times \tilde{C}_w\}}{m}.
\]

We specified $X \times \tilde{C}_w$ rather than $\pi^{-1}(w)$ so that the numerator is forced to only count among those indices $ij$ rather than among all indices $i$.

According to Theorem 7, $\tilde{x}$ is normal with respect to the restriction of $\tilde{T}$ to $X \times Q'_k$. Therefore as $m$ tends to infinity we have that
\[
\frac{\#\{0 \leq i \leq m - 1 : \tilde{T}^i \tilde{x} \in X \times \tilde{C}_w\}}{i_m}
\]
converges to $\tilde{\mu}(X \times \hat{C}_w) / \tilde{\mu}(X \times Q'_k)$. Likewise, since we can write $m$ as

$$\# \{0 \leq i \leq m - 1 : T^i y \in C_w \} = \frac{\tilde{\mu}(X \times \hat{C}_w)}{\tilde{\mu}(X \times (Q'_k \cap F_k))}.$$ 

But, by construction, for any set $E \subset Q_k$, we have that $\tilde{\mu}(X \times E) = \#E / \#Q_k$. Thus, this limsup is equal to

$$\frac{\#\hat{C}_w}{\#(Q'_k \cap F_k)}.$$ 

We want to obtain a crude upper bound on this last fraction. We know that $\hat{C}_w \subseteq F \times \{w\}$, so $\#\hat{C}_w \leq \#F$. Moreover, by the final part of Lemma 8, we know that $\#(Q'_k \cap F_k) \geq b^k$. Thus,

$$\limsup_{m \to \infty} \frac{\# \{0 \leq i \leq m - 1 : T^i y \in C_w \}}{m} \leq \frac{\#F}{b^k}.$$ 

Setting $c = \#F$ completes the proof.

References

1. Agafonov, V.N.: Normal sequences and finite automata. Sov. Math. Doklady 9, 324–325 (1968)
2. Becher, V., Carton, O.: Normal numbers and computer science. In: Berthé, V., Rigo, M. (eds.) Sequences, Groups, and Number Theory, Trends in Mathematics Series. Springer, Birkhäuser (2018)
3. Becher, V., Carton, O., Heiber, P.A.: Normality and automata. J. Comput. Syst. Sci. 81(8), 1592–1613 (2015)
4. Becher, V., Heiber, P.A.: Normal numbers and finite automata. Theor. Comput. Sci. 477, 109–116 (2013)
5. Borel, É.: Les probabilités dénombrables et leurs applications arithmétiques. Rend. Circ. Mat. Palermo 27, 247–271 (1909)
6. Carton, O., Heiber, P.A.: Normality and two-way automata. Inf. Comput. 241, 264–276 (2015)
7. Dai, J., Lathrop, J., Lutz, J., Mayordomo, E.: Finite-state dimension. Theor. Comput. Sci. 310, 1–33 (2004)
8. Furstenberg, H.: Disjointness in ergodic theory, minimal sets, and a problem in Diophantine approximation. Math. Syst. Theory 1, 1–49 (1967)
9. Heersink, B., Vandehey, J.: Continued fraction normality is not preserved along arithmetic progressions. Arch Math. (Basel) 106(4), 363–370 (2016)
10. Kamae, T.: Subsequences of normal sequences. Israel J. Math. 16, 121–149 (1973)
11. Kamae, T., Weiss, B.: Normal numbers and selection rules. Israel J. Math. 21(2-3), 101–110 (1975). Conference on Ergodic Theory and Topological Dynamics (Kibbutz Lavi, 1974)
12. Merkle, W., functions, J.Reimann.: Selection that do not preserve normality. Theory Comput. Syst. 39(5), 685–697 (2006)
13. Moshchevitin, N.G., Shkredov, I.D.: On the pyatetskii-shapiro criterion of normality. Math. Notes 73(3-4), 539–550 (2003)
14. Perrin, D., Pin, J.-É.: Infinite words. Elsevier (2004)
15. Pin, J.-É.: Syntactic semigroups. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, pp. 679–746. Springer (1997)
16. Schnorr, C.P., Stimm, H.: Endliche automaten und zufallsfolgen. Acta Inform. 1, 345–359 (1972)
17. Vandehey, J.: On the joint normality of certain digit expansions. arXiv:1408.0435 (2014)
18. Vandehey, J.: Non-trivial matrix actions preserve normality for continued fractions. Compos. Math. 153(2), 274–293 (2017)
19. Vandehey, J.: Uncanny subsequence selections that generate normal numbers. Unif. Distrib. Theory 12(2), 65–75 (2017)
20. Wall, D.D.: Normal Numbers. ProQuest LLC, Ann Arbor (1950). Thesis (Ph.D.)-University of California, Berkeley
21. Weiss, B.: Normal sequences as collectives. In: Proceedings of Symposium on Topological Dynamics and Ergodic Theory, Univ. of Kentucky (1971)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.