Journal of Computer and System Sciences 81 (2015) 1592–1613

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Normality and automata

Verónica Becher[a,b], Olivier Carton[c], Pablo Ariel Heiber[a,*]

[a] Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Pabellón I, Ciudad Universitaria, (1428) Buenos Aires, Argentina
[b] CONICET, Argentina
[c] Laboratoire d'Informatique Algorithmique: Fondements et Applications, CNRS UMR 7089, Université Paris Diderot – Paris 7, Case 7014, 75205 Paris Cedex 13, France

ARTICLE INFO

Article history:
Received 5 May 2014
Received in revised form 5 March 2015
Accepted 30 April 2015
Available online 10 June 2015

Keywords:
Normal numbers
Finite automata
Non-deterministic automata

ABSTRACT

We prove that finite-state transducers with injective behavior, deterministic or not, real-time or not, with no extra memory or a single counter, cannot compress any normal word. We exhaust all combinations of determinism, real-time, and additional memory in the form of counters or stacks, identifying which models can compress normal words. The case of deterministic push-down transducers is the only one still open. We also present results on the preservation of normality by selection with finite automata. Complementing Agafonov's theorem for prefix selection, we show that suffix selection preserves normality. However, there are simple two-sided selection rules that do not.

1. Introduction

Let us recall the definition of *normality* for real numbers, given by Émile Borel [4] more than one hundred years ago. A real number is normal to an integer base if, in its infinite expansion expressed in that base, all blocks of digits of the same length have the same limiting frequency. Borel proved that almost all real numbers are normal to all integer bases. However, very little is known on how to prove that a given number has the property. The problem of proving normality to just one base does not seem easier, see Bugeaud's book [6]. This is a motivation to investigate different equivalent definitions of normality and different operations that preserve it.

The present work deals with the characterization and the preservation of normality by means of operations done by finite automata. For this, given an integer base b, greater than or equal to 2, we consider the alphabet consisting of the digits $0, 1, \dots, b-1$. We regard the expansion of a real number expressed in base b as a sequence of symbols in this alphabet, or, as we shall call it, an infinite word. This determines the notion of normality for infinite words.

A fundamental theorem relates normality and finite automata: an infinite word is normal to a given alphabet if and only if it cannot be compressed by lossless finite transducers. These are deterministic finite automata with injective input–output behavior. This result was first obtained by joining a theorem by Schnorr and Stimm [17] with a theorem by Dai, Lathrop, Lutz and Mayordomo [10]. Becher and Heiber gave a direct proof [3].

What is the true computational power needed to compress normal words? Of course, each computable normal word is compressible by some Turing machine. For instance, consider Champernowne's construction [9]. Since automata with

[*] Corresponding author.
E-mail addresses: vbecher@dc.uba.ar (V. Becher), Olivier.Carton@liafa.univ-paris-diderot.fr (O. Carton), pheiber@dc.uba.ar (P.A. Heiber).

http://dx.doi.org/10.1016/j.jcss.2015.04.007
0022-0000/© 2015 Elsevier Inc. All rights reserved.

Table 1
Compressibility of normal infinite words by different kinds of transducers.

Finite-state transducer	Deterministic	Non-deterministic	Non-real-time
No extra memory	Not compress (Theorem 2.3)	Not compress (Theorem 4.1)	Not compress (Corollary 4.4)
One counter	Not compress	Not compress	Not compress (Theorem 5.2)
More than one counter	Not compress (Theorem 3.1)	Not compress (Corollary 5.1)	Compress (Turing complete)
One stack	**?**	Compress (Theorem 6.1)	Compress
One stack and one counter	Compress (Theorem 6.2)	Compress	Compress (Turing complete)

enough computational power are equivalent to a Turing machine, to answer the question we ought to start with the most elementary type of automata, which are finite-state automata, and consider different enhancements. Here we analyze deterministic, non-deterministic real-time and non-real-time transition functions; having zero, one, or more counters; with or without a stack.

Whether deterministic or non-deterministic machines have the same computational power is a fundamental question in theoretical computer science. Here we address it with respect to the ability of compressing normal words. Recall that although deterministic and non-deterministic finite automata recognize the same rational sets, deterministic Büchi automata are strictly less expressive than the non-deterministic ones (see Perrin and Pin's book [15]). It is not always possible to determinize a Büchi automaton to recognize the same set of infinite words (it is only possible as a Muller automaton). Furthermore, functions and relations realized by deterministic transducers are proper subclasses of rational functions and relations realized by non-deterministic ones [2].

Here we prove that finite transducers with injective behavior, even non-deterministic non-real-time ones, but with no extra memory or just a single counter, cannot compress any normal infinite word. Adding memory yields compressibility results: there are non-deterministic non-real-time transducers with more than one counter that compress some normal infinite words. Also there are non-deterministic real-time transducers with a stack that can do it.

Table 1 summarizes the results we obtain about compressibility of normal infinite words by different kinds of transducers. The columns represent different levels of restrictions on the transitions. The first column represents determinism, that is, there is exactly one transition leaving a given state by reading a given symbol. The second column represents non-determinism, there are several transitions leaving a given state by reading the same symbol. The restriction represented in the third column adds the possibility of also having transitions that do not read any symbol (usually called λ-transitions). The rows of the table represent different memory models. In all cases there is bounded memory represented by states. Each row details possible additions of counters or stacks. The realized relation is assumed to be bounded-to-one. The case of a deterministic transducers with a single stack remains open:

Open question. Can any deterministic push-down transducer compress a normal word?

In the present paper, we consider transducers that process the input word from left to right without coming back. These are called one-way transducers. In contrast, two-way transducers can move their reading head back and forth. They have been investigated with regards to normality by Carton and Heiber in [7] where it is shown that these more powerful transducers still cannot compress normal words.

We also obtain new results on the preservation of normality by selection. A celebrated theorem by Agafonov describes an operation by a finite automaton that selects symbols from an infinite word by looking at its prefixes and by recognizing those that belong to a rational set. In case the word is normal, the word obtained by the selected symbols is normal as well. Agafonov published it in 1968 [1], but unfortunately the proof there depends on work only available in the Russian literature. Schnorr and Stimm [17] proved a generalization of the theorem. M. O'Connor [14] gave another proof of Agafonov's result using automata predictors, and Broglio and Liardet [5] generalized it to arbitrary alphabets. Becher and Heiber [3] wrote an alternative proof using the characterization of normality in terms of incompressibility. It is known that Agafonov's theorem fails for slightly more powerful selection. Merkle and Reimann [12] showed that normality is preserved neither by deterministic one-counter sets (recognized by deterministic one-counter automata) nor by linear sets (recognized by one-turn pushdown automata).

Here we complement Agafonov's theorem and we show that selection based on suffixes, as opposed to prefixes, also preserves normality. However, there are simple two-sided selection rules that do not preserve normality and we exhibit one. These results are proved in Theorems 7.2 and 7.3.

1.1. Notation

We write $\mathbb{Z}$ for the set of all integers. An alphabet is a finite set with at least two symbols. A word over alphabet A is a sequence of elements from A. A^ℓ is the set of words of ℓ symbols from A, $A^{<\ell} = \bigcup_{\ell' < \ell} A^{\ell'}$ is the set of words of less than ℓ symbols from A, $A^* = \bigcup_{\ell \geq 0} A^\ell$ is the set of all finite words over A and A^ω is the set of all infinite words over A. We mostly use upper-case letters to denote sets or other complex objects, lower-case letters a, b, c to denote alphabet symbols, u, v, w to denote finite words and x, y, z to denote infinite words. The empty word is denoted by λ, the length of a word u is $|u|$ and if u and v are words, uv is the concatenation of u and then v. Similarly, ux is the concatenation of word u and then an infinite word x. We write $x \upharpoonright \ell$ for the prefix of length ℓ of x and $x \upharpoonleft \ell$ for the suffix of x that results on removing the first ℓ symbols from it. As usual, we say that a set of words is rational if it can be recognized by some finite automaton. For any finite set S we denote its cardinality with $|S|$. We write log for the logarithm in base 2.

1.2. Normality

We consider the definition of normality directly on infinite words over an alphabet A. See the books [6,11] for a thorough presentation of the material.

Definition. Let ℓ be a positive integer. An infinite word $x \in A^\omega$ is *simply normal* to word length ℓ if $x = v_1 v_2 v_3 \cdots$ where for each $i \geq 1$, $|v_i| = \ell$, and for every $u \in A^\ell$,

$$\lim_{n \to \infty} \frac{|\{i : 1 \leq i \leq n, u = v_i\}|}{n} = |A|^{-\ell}.$$

An infinite word x is *normal* if it is simply normal to every word length.

The concept of normality can be equivalently characterized according to the following theorem. We will use this characterization in Section 7.

Theorem 1.1. *(See Theorem 4.2 [6], originally proved by Pillai between 1939 and 1940.) An infinite word $x = a_1 a_2 a_3 \cdots \in A^\omega$ is* normal *if, and only if, for every $u \in A^*$,*

$$\lim_{n \to \infty} \frac{|\{i : 1 \leq i \leq n - |u| + 1, u = a_i a_{i+1} \cdots a_{i+|u|-1}\}|}{n} = |A|^{-|u|}.$$

2. Deterministic transducers

We recall the standard definition of a deterministic transducer and fix notation. A transducer is an automaton equipped with an output tape. The execution of each transition consumes at most one symbol from the input and produces a word on the output. Thus, the transducer outputs the concatenation of all the words output by the executed transitions.

We first introduce transducers where the underlying automaton is deterministic. These transducers are also called sequential in the literature [16].

Definition. A deterministic *transducer* is a tuple $T = \langle Q, A, B, \delta, q_0 \rangle$, where

- Q 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
- $q_0 \in Q$ is the starting state.

The transducer T processes infinite words over A: if at state p symbol a is processed, T moves to state q and outputs a word v where $\langle v, q \rangle = \delta(p, a)$. In this case, we write $p \xrightarrow{a|v} q$. Notice that v can be empty.

A *finite run* of the transducer is a finite sequence of consecutive transitions

$$p_0 \xrightarrow{a_1|v_1} p_1 \xrightarrow{a_2|v_2} p_2 \cdots p_{n-1} \xrightarrow{a_n|v_n} p_n$$

and we write $p_0 \xrightarrow{u|v} p_n$ where $u = a_1 a_2 \cdots a_n$ and $v = v_1 v_2 \cdots v_n$.

An *infinite run* of the transducer is a sequence of consecutive transitions

$$p_0 \xrightarrow{a_1|v_1} p_1 \xrightarrow{a_2|v_2} p_2 \xrightarrow{a_3|v_3} p_3 \cdots$$

and we write $p_0 \xrightarrow{x|y} \infty$ where $x = a_1 a_2 a_3 \cdots$ and $y = v_1 v_2 v_3 \cdots$. An infinite run is *accepting* if $p_0 = q_0$. This is the Büchi acceptance condition where all states are accepting. We write $T(x)$ to refer to the word such that $q_0 \xrightarrow{x|T(x)} \infty$.

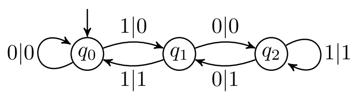

Fig. 1. A transducer for the division by 3 in base 2.

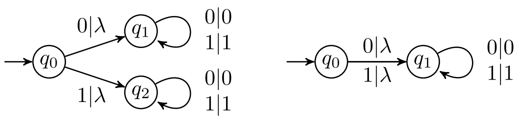

Fig. 2. A lossless and a 2-to-one transducer.

When it is clear from context we may omit whether a run is finite or infinite. Notice that in this definition $T(x)$ is not required to be an infinite word. However, in the next theorems we impose a condition on the transducers that ensures that the output is infinite. Hereafter a transducer is a deterministic transducer unless indicated otherwise.

The transducer pictured in Fig. 1 realizes the following function from binary words to binary words. If the input x is the binary expansion of some real number α in the unit interval, then the output is the binary expansion of $\alpha/3$. This function is not one-to-one since dyadic numbers have two binary expansions. The two binary expansions $01111\cdots$ and $10000\cdots$ of $1/2$ are mapped to the unique binary expansion $0010101\cdots$ of $1/6$.

We say that a state q is *reachable* if there is a finite run from the starting state to q.

Definition. Let $T = \langle Q, A, B, \delta, q_0\rangle$ be a transducer.

1. T is *one-to-one* if the function $x \mapsto T(x)$ is one-to-one.
2. T is *lossless* if for every pair of different words u_1 and u_2, it is not true that $q_0 \xrightarrow{u_1|v} p$ and $q_0 \xrightarrow{u_2|v} p$ for some word v and state p.
3. T is *bounded-to-one* if the function $x \mapsto T(x)$ is bounded-to-one.

Being lossless is a property defined on the structure of the transducer whereas being one-to-one or bounded-to-one is defined on the realized function. Observe that the same function can be realized by a lossless or a non-lossless transducer as shown by the following example.

Consider the two transducers pictured in Fig. 2. They both realize the shift function that maps each infinite word $x = a_1a_2a_3\cdots$ to the infinite word $y = a_2a_3a_4\cdots$ obtained by removing its first symbol. The first one is lossless whereas the second one is not. However they are both 2-to-one, and of course, neither is one-to-one.

Proposition 2.1. *Every one-to-one transducer is lossless. Every lossless transducer is bounded-to-one.*

Proof. Let $T = \langle Q, A, B, \delta, q_0\rangle$ be a transducer. For the first implication, assume T is not lossless. Then there are different words u_1 and u_2, a word v and a state p such that $q_0 \xrightarrow{u_1|v} p$ and $q_0 \xrightarrow{u_2|v} p$. Let x be a non-periodic infinite word, then it is clear that $u_1x \neq u_2x$ but $q_0 \xrightarrow{u_1|v} p \xrightarrow{x|y} \infty$ and $q_0 \xrightarrow{u_2|v} p \xrightarrow{x|y} \infty$ for some y, thus $T(u_1x) = T(u_2x) = vy$, so T is not one-to-one.

For the second implication, assume T is lossless, and thus, there is no cyclic run $p \xrightarrow{u|\lambda} p$ with u non-empty and p a reachable state. Therefore $T(x)$ is infinite for all infinite words x. Let $x_1, \ldots, x_n$ be different infinite words that yield the same output $y = T(x_i)$ for $i = 1, \ldots, n$. Let ℓ be the minimum length such that the prefixes $u_i = x_i \upharpoonright \ell$ are mutually distinct. Let

$$k = \max\{|w| : \exists p \in Q, \exists i, 1 \le i \le n, q_0 \xrightarrow{u_i|w} p\}.$$

Let v_i be the shortest prefix of x_i such that for some word w_i with $|w_i| > k$ and state p_i, $q_0 \xrightarrow{v_i|w_i} p_i$. Let

$$m = \max\{|w| : \exists a \in A, \exists p, q \in Q, p \xrightarrow{a|w} q\}$$

be the length of the longest output of a transition of T. Since each transition only adds at most m symbols to the output, $|w_i| > k + m$ would imply v_i is not shortest. Therefore, $k + 1 \le |w_i| \le k + m$. By definition, for each i, $|v_i| > \ell$, hence the v_i are pairwise different. Since T is lossless, the tuples $\delta(q_0, v_i)$ are pairwise different and they are included in the set $Q \times \{y \upharpoonright j : k + 1 \le j \le k + m\}$. Therefore, there are at most $|Q|m$ such tuples, so $n \le |Q|m$ and T is $(|Q|m)$-to-one. □

Definition. An infinite word $x = a_1a_2a_3\cdots$ is *compressible* by a transducer if its accepting run $q_0 \xrightarrow{a_1|v_1} q_1 \xrightarrow{a_2|v_2} q_2 \xrightarrow{a_3|v_3} q_3 \cdots$ satisfies

$$\liminf_{n\to\infty} \frac{|v_1v_2\cdots v_n|}{n} \frac{\log|B|}{\log|A|} < 1.$$

Notice that the factor $\log|B|/\log|A|$ in the definition above just accounts for the necessary recoding from one alphabet to another one. Hereafter, to ease the presentation of the proofs we assume that $|A|=|B|$. The generalization is straightforward: if the lengths of words over B are multiplied by the factor $\log|B|/\log|A|$ whenever they are compared to lengths of words over A, all the proofs hold for the general case.

Lemma 2.2. *Let ℓ be a positive integer, and let $u_1, u_2, u_3, \ldots$ be words of length ℓ over the alphabet A such that $u_1u_2u_3\cdots$ is simply normal to word length ℓ. Let*

$$C_0 \xrightarrow{u_1|v_1} C_1 \xrightarrow{u_2|v_2} C_2 \xrightarrow{u_3|v_3} C_3 \cdots$$

be a run where each C_i is a configuration of some kind of transducer. Assume there is a real $\varepsilon > 0$ and a set $U \subseteq A^\ell$ of at least $(1-\varepsilon)|A|^\ell$ words such that $u_i \in U$ implies $|v_i| \geq \ell(1-\varepsilon)$. Then,

$$\liminf_{n\to\infty} \frac{|v_1v_2\cdots v_n|}{n\ell} \geq (1-\varepsilon)^3.$$

Proof. Assume words u_i as in the hypothesis. By definition of normality to word length ℓ, let n_0 be such that for every $u \in A^\ell$ and for every $n \geq n_0$,

$$|\{i : 1 \leq i \leq n, u_i = u\}| \geq n|A|^{-\ell}(1-\varepsilon).$$

Then, for every $n \geq n_0$,

$$\begin{aligned}
|v_1v_2\cdots v_n| &= \sum_{i=1}^{n} |v_i| \\
&\geq \sum_{1\leq i\leq n, u_i\in U} |v_i| \\
&\geq \sum_{1\leq i\leq n, u_i\in U} \ell(1-\varepsilon) \\
&\geq n|A|^{-\ell}(1-\varepsilon)\sum_{u\in U} \ell(1-\varepsilon) \\
&\geq n|A|^{-\ell}(1-\varepsilon)(1-\varepsilon)|A|^\ell \ell(1-\varepsilon) \\
&\geq (1-\varepsilon)^3 n\ell. \qquad \square
\end{aligned}$$

The next theorem extends the known characterization of normality as incompressibility by lossless finite transducers to bounded-to-one. This proof is a slight generalization of the one given by Becher and Heiber [3].

Theorem 2.3. *No normal infinite word is compressible by a bounded-to-one transducer.*

Proof. Fix a normal infinite word $x = a_1a_2a_3\cdots$, a bounded-to-one transducer $T = \langle Q, A, B, \delta, q_0\rangle$ with only reachable states, a real $\varepsilon > 0$ and the accepting run $q_0 \xrightarrow{a_1|v_1} q_1 \xrightarrow{a_2|v_2} q_2 \xrightarrow{a_3|v_3} q_3 \cdots$. It suffices to show that there is ℓ and U such that Lemma 2.2 applies to this arbitrary choice of T and ε. For each word $u \in A^*$ let

$$h_u = \min\{|v| : \exists p, q \in Q, p \xrightarrow{u|v} q\}$$

be the minimum number of symbols that the processing of u can contribute to the output. Let

$$U_\ell = \{u \in A^\ell : h_u \geq (1-\varepsilon)\ell\}$$

be the set of words of length ℓ with relatively large contribution to the output. Let t be such that T is t-to-one. For each length ℓ, pair of states $p, q \in Q$ and word v, consider the set

$$U' = \{u \in A^\ell : p \xrightarrow{u|v} q\}.$$

Since p is reachable, let u_0, v_0 be such that $q_0 \xrightarrow{u_0|v_0} p$. Thus, for different $u_1, u_2 \in U'$, $q_0 \xrightarrow{u_0u_1|v_0v} q$ and $q_0 \xrightarrow{u_0u_2|v_0v} q$, therefore $T(u_0u_1x) = T(u_0u_2x)$ for any x, and by the definition of t, $|U'| \leq t$. By bounding the cardinality of the sets U', we can bound the complement of U_ℓ for each ℓ.

$$\begin{aligned} \forall p, q \in Q, \forall v \in B^*, \quad & |\{u \in A^\ell : p \xrightarrow{u|v} q\}| && \le t \\ \forall v \in B^*, & |\{u : \exists p, q \in Q,\ p \xrightarrow{u|v} q\}| \le |Q|^2 t \\ & |\{u : h_u < (1-\varepsilon)\ell\}| && \le |Q|^2 t |B|^{(1-\varepsilon)\ell+1}. \end{aligned}$$

Thus, $|U_\ell| \ge |A|^\ell - |Q|^2 t |B|^{(1-\varepsilon)\ell+1}$. Fix ℓ such that $|U_\ell| > |A|^\ell(1-\varepsilon)$, which is possible because the subtracted term in the lower bound of the last inequality is $o(|A|^\ell)$ (recall $|A| = |B|$), and take $U = U_\ell$. By construction, the only run of T over x fulfills the hypothesis of Lemma 2.2 using states as configurations. The application of the lemma finishes the proof. □

3. Counter transducers

We consider deterministic transducers augmented with a fixed number of *counters*. Each counter contains an integer value. The execution of each transition reads exactly one input symbol, checks each counter for being zero or non-zero, and increments or decrements each counter by some amount. Thus, a counter can only increase or decrease by a bounded amount when processing a symbol of the input. Notice we are assuming, by default, that the transducers process in real-time, which means that each transition necessarily consumes a symbol of the input. We will consider transducers that are not real-time in Section 4.2.

We introduce some notation for tuples, to be used in the rest of the document. When the range 1 to k is clear from the context, we write $\overline{m}$ for the k-tuple $\langle m_1, \dots, m_k\rangle$ and $\bar{0}$ for the k-tuple $\langle 0, \dots, 0\rangle$.

Definition. A (deterministic) *k-counter transducer* is a tuple $T = \langle Q, A, B, \delta, q_0\rangle$, where

- Q is a finite set of states,
- A and B are the input and output alphabets, respectively,
- $\delta : Q \times \{\text{true}, \text{false}\}^k \times A \to B^* \times Q \times \mathbb{Z}^k$ is the transition function,
- $q_0 \in Q$ is the starting state.

The transducer T processes infinite words over A: if at state p with values $\overline{m}$ in the counters, symbol a is processed, T moves to state q, stores value $\bar{n} = \overline{m} + \bar{d}$ in the counters (namely, $m_i = n_i + d_i$ in counter i) and outputs v where $\langle v, q, \bar{d}\rangle = \delta(p, \bar{c}, a)$ and each c_i indicates whether $m_i = 0$ or not. Such a transition of the transducer is denoted by $\langle p, \overline{m}\rangle \xrightarrow{a|v} \langle q, \bar{n}\rangle$.

A *finite run* of the transducer is a finite sequence of consecutive transitions

$$\langle p_0, \overline{m}_0\rangle \xrightarrow{a_1|v_1} \langle p_1, \overline{m}_1\rangle \xrightarrow{a_2|v_2} \langle p_2, \overline{m}_2\rangle \cdots \langle p_{n-1}, \overline{m}_{n-1}\rangle \xrightarrow{a_n|v_n} \langle p_n, \overline{m}_n\rangle$$

and we write $\langle p_0, \overline{m}_0\rangle \xrightarrow{u|v} \langle p_n, \overline{m}_n\rangle$ where $u = a_1 a_2 \cdots a_n$ and $v = v_1 v_2 \cdots v_n$.

An *infinite run* of the transducer is a sequence of consecutive transitions

$$\langle p_0, \overline{m}_0\rangle \xrightarrow{a_1|v_1} \langle p_1, \overline{m}_1\rangle \xrightarrow{a_2|v_2} \langle p_2, \overline{m}_2\rangle \xrightarrow{a_3|v_3} \langle p_3, \overline{m}_3\rangle \cdots$$

and we write $\langle p_0, \overline{m}_0\rangle \xrightarrow{x|y} \infty$ where $x = a_1 a_2 a_3 \cdots$ and $y = v_1 v_2 v_3 \cdots$. An infinite run is accepting if $p_0 = q_0$ and all initial values of the counters are 0, namely, $\overline{m}_0 = \bar{0}$. This is the Büchi acceptance condition where all states are accepting. We write $T(x)$ to refer to the word such that $\langle q_0, \bar{0}\rangle \xrightarrow{x|T(x)} \infty$.

Notice that for given p, $\overline{m}$ and u there is a unique choice of v, q and $\bar{n}$ such that $\langle p, \overline{m}\rangle \xrightarrow{u|v} \langle q, \bar{n}\rangle$. A configuration $\langle q, \overline{m}\rangle$ is reachable if there is a finite run from the starting configuration $\langle q_0, \bar{0}\rangle$ to $\langle q, \overline{m}\rangle$. Extending the definition of bounded-to-one and compressibility to counter transducers is straightforward.

Definition. A counter transducer T is *bounded-to-one* if the function $x \mapsto T(x)$ is bounded-to-one.

Definition. An infinite word $x = a_1 a_2 a_3 \cdots$ is *compressible* by a counter transducer if its accepting run $\langle q_0, \bar{0}\rangle \xrightarrow{a_1|v_1} \langle q_1, \overline{m}_1\rangle \xrightarrow{a_2|v_2} \langle q_2, \overline{m}_2\rangle \xrightarrow{a_3|v_3} \langle q_3, \overline{m}_3\rangle \cdots$ satisfies

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

Theorem 3.1. *No normal infinite word is compressible by a bounded-to-one counter transducer.*

Before proving Theorem 3.1 we need to introduce some technical tools.

Definition. Let $T = \langle Q, A, B, \delta, q_0\rangle$ be a counter transducer. We define D_T as the maximum absolute value of a counter increment or decrement in a transition in δ,

$$D_T = \max\{|d_i| : 1 \le i \le k, \exists p, q \in Q\ \exists a \in A\ \exists v \in B^*\ \exists \bar{c} \in \{\text{true}, \text{false}\}^k\ \langle v, q, \bar{d}\rangle = \delta(p, \bar{c}, a)\}.$$

All runs with values of the counters far from 0 have a similar behavior, because if a counter does not become zero during the run, then its value has no impact. We formalize this with the concept of *template*.

Definition. For each positive integer L let $\pi_L : \mathbb{Z} \to [-L, L] \cup \{-\infty, +\infty\}$ be the function that identifies each integer in $(-\infty, -L)$ with $-\infty$ and each integer in $(L, +\infty)$ with $+\infty$,

$$\pi_L(n) = \begin{cases} -\infty & \text{if } n < -L \\ n & \text{if } -L \le n \le L \\ +\infty & \text{if } n > -L \end{cases}$$

The function π_L can be extended component-wise to tuples of integers by setting $\pi_L(\overline{m}) = \bar{n}$ where $n_i = \pi_L(m_i)$.

For a positive integer L, an L-*template* for a k-counter transducer is a triple $\langle \overline{M}, \overline{N}, \overline{O} \rangle$ where each $\overline{M}, \overline{N}, \overline{O}$ is a k-tuple in $([-L, L] \cup \{-\infty, +\infty\})^k$. We say that a run $\langle p, \overline{m} \rangle \xrightarrow{u|v} \langle q, \bar{n} \rangle$ of the transducer *complies* only to one L-template; namely, $\langle \pi_L(\overline{m}), \pi_L(\bar{n}), \pi_L(\overline{m} - \bar{n}) \rangle$.

Observe that there are exactly $(2L+3)^{3k}$ L-templates of a k-counter transducer.

Lemma 3.2. *Let u and u' be two words of length ℓ. Let $\langle p, \overline{m} \rangle \xrightarrow{u|v} \langle q, \bar{n} \rangle$ and $\langle p, \overline{m}' \rangle \xrightarrow{u'|v} \langle q, \bar{n}' \rangle$ be two runs of the same k-counter transducer T that comply to the same (ℓD_T)-template $\langle \overline{M}, \overline{N}, \overline{O} \rangle$. Then, there exists a run $\langle p, \overline{m}' \rangle \xrightarrow{u|v} \langle q, \bar{n}' \rangle$ of the same transducer.*

Proof. First notice that for each counter i, $m_i - n_i$ is bounded by ℓD_T because u has length ℓ and at each step of the run a counter cannot increase or decrease more than D_T. Since both runs comply to the same (ℓD_T)-template and the difference in counters is within the interval $[-\ell D_T, \ell D_T]$, the difference in counters on both runs must coincide. This is true for each counter, so $\overline{m} - \bar{n} = \overline{m}' - \bar{n}'$, which implies $\overline{m} - \overline{m}' = \bar{n} - \bar{n}'$. Let us write the run over $u = a_1 a_2 \cdots a_\ell$ explicitly:

$$\langle p, \overline{m} \rangle = \langle p_0, \overline{m}_0 \rangle \xrightarrow{a_1|v_1} \langle p_1, \overline{m}_1 \rangle \xrightarrow{a_2|v_2} \cdots \xrightarrow{a_\ell|v_\ell} \langle p_\ell, \overline{m}_\ell \rangle = \langle q, \bar{n} \rangle$$

where $v = v_1 v_2 \cdots v_\ell$. Consider the tuple $\bar{d} = \overline{m} - \overline{m}' = \bar{n} - \bar{n}'$. We need only to prove that the following run over u,

$$\langle p, \overline{m}' \rangle = \langle p_0, \overline{m}_0 - \bar{d} \rangle \xrightarrow{a_1|v_1} \langle p_1, \overline{m}_1 - \bar{d} \rangle \xrightarrow{a_2|v_2} \cdots \xrightarrow{a_\ell|v_\ell} \langle p_\ell, \overline{m}_\ell - \bar{d} \rangle = \langle q, \bar{n}' \rangle,$$

is a valid run. Let us show that in configuration $\langle p_j, \overline{m}_j \rangle$ counter i is zero if and only if it is zero in configuration $\langle p_j, \overline{m}_j - \bar{d} \rangle$. That is, $m_{j,i}$ is zero if and only if $m_{j,i} - d_i$ is zero. Then, validity of each step follows from validity of the corresponding step in the original run. If d_i is zero, the requirement follows immediately. If d_i is non-zero, then $m_i \neq m_i'$, and therefore m_i and m_i' are both greater than ℓD_T or both smaller than ℓD_T. Assume $m_i, m_i' > \ell D_T$. Since $m_{j,i} \ge m_{0,i} - D_T j = m_i - D_T j$ and $j \le \ell$, $m_{j,i}$ is positive. And $m_{j,i} - d_i = m_{j,i} - m_i + m_i' = m_i' - (m_i - m_{j,i}) \ge m_i' - D_T j$ is also positive. If, on the other hand, $m_i, m_i' < \ell D_T$, it follows by symmetry that both $m_{j,i}$ and $m_{j,i} - d_i$ are negative. □

Proof of Theorem 3.1. Fix a normal infinite word x, a bounded-to-one k-counter transducer $T = \langle Q, A, B, \delta, q_0 \rangle$, a real $\varepsilon > 0$ and the accepting run

$$\langle q_0, \bar{0} \rangle \xrightarrow{a_1|v_1} \langle q_1, \overline{m}_1 \rangle \xrightarrow{a_2|v_2} \langle q_2, \overline{m}_2 \rangle \xrightarrow{a_3|v_3} \cdots.$$

It suffices to show that there is ℓ and U such that Lemma 2.2 applies to this arbitrary choice of T and ε. For each word $u \in A^*$ let

$$h_u = \min\{|v| : \exists p, q \in Q \ \exists \overline{m}, \bar{n} \in \mathbb{Z}^k, \langle p, \overline{m} \rangle \xrightarrow{u|v} \langle q, \bar{n} \rangle\}$$

be the minimum number of symbols that the processing of u can contribute to the output. Let

$$U_\ell = \{u \in A^\ell : h_u \ge (1 - \varepsilon)\ell\}$$

be the set of words of length ℓ with relatively large contribution to the output. Let t be such that T is t-to-one. For each pair of states $p, q \in Q$, length ℓ, word v and (ℓD_T)-template τ, consider the following set

$$U' = \{u \in A^\ell : \exists \overline{m}, \bar{n} \in \mathbb{Z}^k, \langle p, \overline{m} \rangle \xrightarrow{u|v} \langle q, \bar{n} \rangle \text{ complies to } \tau, \langle p, \overline{m} \rangle \text{ is reachable}\}.$$

Let $u_1, u_2, \ldots, u_n$ be the n different words in U'. Since each u_i is in U', let $\overline{m}_1$ and $\bar{n}_1$ be such that $\langle p, \overline{m}_1 \rangle \xrightarrow{u_1|v} \langle q, \bar{n}_1 \rangle$ complies to τ and $\langle p, \overline{m}_1 \rangle$ is reachable. By Lemma 3.2 with $u = u_i$ and $u' = u_1$, for each i there exist runs $\langle p, \overline{m}_1 \rangle \xrightarrow{u_i|v} \langle q, \bar{n}_1 \rangle$. Since $\langle p, \overline{m}_1 \rangle$ is reachable, let u_0, v_0 be such that $\langle q_0, \bar{0} \rangle \xrightarrow{u_0|v_0} \langle p, \overline{m}_1 \rangle$. Therefore, there is an accepting run $\langle q_0, \bar{0} \rangle \xrightarrow{u_0|v_0} \langle p, \overline{m}_1 \rangle \xrightarrow{u_i|v} \langle q, \bar{n}_1 \rangle \xrightarrow{x|y} \infty$ which shows $T(u_0 u_i x) = v_0 v y$ for each i. Therefore, by definition of t, $n \le t$ and $|U'| \le t$. Now we can continue the proof as in the case with no counters, given in Theorem 2.3.

$$|\{u \in A^\ell : \exists \overline{m}, \bar{n} \in \mathbb{Z}^k, \langle p, \overline{m} \rangle \xrightarrow{u|v} \langle q, \bar{n} \rangle \text{ complies to } \tau, \langle p, \overline{m} \rangle \text{ is reachable}\}| \le t$$

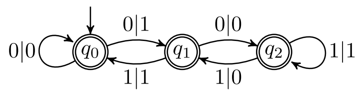

Fig. 3. A transducer for the multiplication by 3 in base 2.

So,

$$|\{u \in A^\ell : \exists p, q \in Q\ \exists\tau\ \exists \overline{m}, \overline{n} \in \mathbb{Z}^k \langle p, \overline{m}\rangle \xrightarrow{u|v} \langle q, \overline{n}\rangle \text{ complies to } \tau, \langle p, \overline{m}\rangle \text{ is reachable}\}|$$

is at most $|Q|^2(2\ell D_T + 3)^{3k}t$. Then,

$$|\{u : |u| = \ell, h_u < (1-\varepsilon)\ell\}| \le |Q|^2(2\ell D_T + 3)^{3k}t|B|^{(1-\varepsilon)\ell+1}$$

and

$$|U_\ell| \ge |A|^\ell - |Q|^2(2\ell D_T + 3)^{3k}t|B|^{(1-\varepsilon)\ell+1}.$$

Fix ℓ such that $|U_\ell| > |A|^\ell(1-\varepsilon)$ and apply Lemma 2.2 with $U = U_\ell$ to the considered run. This completes the proof. □

4. Non-deterministic transducers

4.1. Non-deterministic real-time transducers

We consider *non-deterministic transducers*. We focus first on transducers that operate in real-time, that is, they process exactly one input alphabet symbol per transition.

Definition. A *non-deterministic transducer* is a tuple $T = \langle Q, A, B, \delta, q_0, F\rangle$, where

- Q is a finite set of states,
- A and B are the input and output alphabets, respectively,
- $\delta \subset Q \times A \times B^* \times Q$ is a finite transition relation,
- $q_0 \in Q$ is the starting state.
- $F \subseteq Q$ is the set of accepting states,

T processes infinite words over A: if at state p symbol a is processed, T may move to a state q and output v whenever $\delta(p, a, v, q)$. In this case, we write $p \xrightarrow{a|v} q$. Finite and infinite runs are defined as in the case of deterministic transducers, and an infinite run is accepting if it starts at q_0 and visits infinitely often accepting states. This is the classical Büchi acceptance condition. We write $T(x, y)$ whenever there is an accepting run $q_0 \xrightarrow{x|y} \infty$.

Note that we only consider transducers with a Büchi acceptance condition since any transducer with a stronger acceptance condition like the Muller one is equivalent to a non-deterministic transducer with a Büchi acceptance condition.

Definition. A non-deterministic transducer T is *bounded-to-one* if the function $y \mapsto |\{x : T(x, y)\}|$ is bounded.

Exchanging the input and the output of each transition of the transducer of Fig. 1 yields the transducer of Fig. 3. This transducer is non-deterministic. If the input x is the infinite binary expansion in base 2 of some real number $\alpha < 1/3$, then the output is the binary expansion of 3α.

Definition. An infinite word $x = a_1a_2a_3\cdots$ is *compressible* by a non-deterministic transducer 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

$$\liminf_{n\to\infty} \frac{|v_1v_2\cdots v_n| \log|B|}{n \log|A|} < 1.$$

Theorem 4.1. *No normal infinite word is compressible by a bounded-to-one non-deterministic transducer.*

Proof. Fix a normal infinite word $x = a_1a_2a_3\cdots$, a real $\varepsilon > 0$, a bounded-to-one non-deterministic transducer $T = \langle Q, A, B, \delta, q_0, F\rangle$, and an accepting run $q_0 \xrightarrow{a_1|v_1} q_1 \xrightarrow{a_2|v_2} q_2 \xrightarrow{a_3|v_3} q_3 \cdots$. It suffices to show that there is ℓ and U such that Lemma 2.2 applies to this arbitrary choice of ε, T and accepting run. For each word $u \in A^*$ let

$$h_u = \min\{|v| : \exists i, j, 0 \le i \le j, q_i \xrightarrow{u|v} q_j\}$$

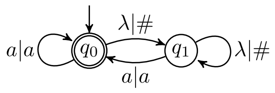

Fig. 4. A non-real-time transducer that inserts dummy symbols #.

be the minimum number of symbols that the processing of u can contribute to the output in the run we fixed. Let

$$U_\ell = \{u \in A^\ell : h_u \geq (1-\varepsilon)\ell\}$$

be the set of words of length ℓ with relatively large contribution to the output. Let t be such that T is t-to-one. For each length ℓ, pair of states p, q that appear in the run, and for each word v, consider the set

$$U' = \{u \in A^\ell : p \xrightarrow{u|v} q\}.$$

Since p and q appear in the run, let $q_0 \xrightarrow{u_0|v_0} p$ be a prefix of the run and $q \xrightarrow{x_0|y_0} \infty$ be a suffix of the run. This implies $q \xrightarrow{x_0|y_0} \infty$ goes infinitely often through an accepting state. Thus, for different $u_1, u_2 \in U'$, there are accepting runs $q_0 \xrightarrow{u_0u_1x_0|v_0vy_0} \infty$ and $q_0 \xrightarrow{u_0u_2x_0|v_0vy_0} \infty$, from which it follows that $T(u_0u_1x_0, v_0vy_0)$ and $T(u_0u_2x_0, v_0vy_0)$. Therefore, by definition of t, $|U'| \leq t$. Now we can continue the proof as in the deterministic case, given in Theorem 2.3,

$$|\{u \in A^\ell : p \xrightarrow{u|v} q\}| \leq t.$$

Thus,

$$|U_\ell| \geq |A|^\ell - |Q|^2 t |B|^{(1-\varepsilon)\ell+1}.$$

Fix ℓ such that $|U_\ell| > |A|^\ell(1-\varepsilon)$ and apply Lemma 2.2 with $U = U_\ell$ to the considered run. This completes the proof. □

4.2. Non-deterministic non-real-time transducers

A *non-real-time* (necessarily non-deterministic) transducer $T = \langle Q, A, B, \delta, q_0, F\rangle$ is identical to a non-deterministic transducer, with the exception that the transition relation δ is a finite subset of $Q \times (A \cup \{\lambda\}) \times B^* \times Q$ instead of $Q \times A \times B^* \times Q$. This allows the transducer to execute without processing a symbol of the input word. If the machine is at state p and $\delta(p, \lambda, v, q)$ holds, the machine can move to state q and output v without processing the next input symbol. We extend the definition and notation of runs and accepting runs to this case, and write the relation T in the same way as for non-deterministic transducers.

We give here an example of a relation T that can be realized by a non-real-time transducer but that cannot be realized by a real-time one. Let A be any alphabet and let B be the alphabet $A \cup \{\#\}$ obtained by adding to A a new dummy symbol $\# \notin A$. In this machine, $T(x, y)$ if and only if y is obtained by inserting dummy symbols in x. This relation T is clearly one-to-one since x is recovered from y by removing all the symbols #. It is easy to see that this relation cannot be realized by a non-deterministic real-time transducer. On the other hand, Fig. 4 shows a non-real-time transducer that realizes it. Note that state q_0 is accepting. This forces the transducer to copy x entirely. Otherwise, the word y could end with a tail $\#^\omega$ and the transducer would not be bounded-to-one.

Definition. An infinite word x over A is *compressible* by a non-real-time transducer if it has an accepting run $q_0 \xrightarrow{b_1|v_1} q_1 \xrightarrow{b_2|v_2} q_2 \xrightarrow{b_3|v_3} q_3 \cdots$, where each $b_i \in A \cup \{\lambda\}$ and $x = b_1b_2b_3\cdots$ satisfying

$$\liminf_{n\to\infty} \frac{|v_1v_2\cdots v_n| \log|B|}{|b_1b_2\cdots b_n| \log|A|} < 1.$$

In this definition, if each b_i were different from λ then $|b_1b_2\cdots b_n| = n$ and the definition would coincide with compressibility for real-time transducers.

Real-time transducers always read the entire input. However, non-real-time transducers may loop forever using transitions without input. The next lemma shows that this cannot happen for accepting runs of bounded-to-one non-real-time transducers.

Lemma 4.2. *Every accepting run of a bounded-to-one non-real-time transducer reads the entire input.*

Proof. Fix the transducer T. If an accepting run over input wx with output y reads only w, then the same run is an accepting run of any input wx', thus, making $T(wx', y)$ true for any x', which contradicts the bounded-to-one condition. □

Theorem 4.3. *For every bounded-to-one non-real-time transducer there exists a non-deterministic (real-time) transducer that compresses exactly the same infinite words.*

Proof. Let $T = \langle Q, A, B, \delta, q_0, F\rangle$ be a non-real-time transducer. Let us show that for any run of T consisting of n transitions $p_0 \xrightarrow{\lambda|v_1} p_1 \xrightarrow{\lambda|v_2} p_2 \cdots p_{n-1} \xrightarrow{\lambda|v_n} p_n$ there exists a run $p_0 \xrightarrow{\lambda|v} p_n$ consisting of at most $2|Q|$ transitions, such that $|v| \le |v_1 v_2 \cdots v_n|$ and it visits an accepting state if and only if the original one does. If $n > 2|Q|$, there is a state q visited three times in the run, so we can write the run as $p_0 \xrightarrow{\lambda|w_1} q \xrightarrow{\lambda|w_2} q \xrightarrow{\lambda|w_3} q \xrightarrow{\lambda|w_4} p_n$ where $v_1 v_2 \cdots v_n = w_1 w_2 w_3 w_4$. Notice that if $q = p_0$ and/or $q = p_n$ the first and/or last subruns may be empty. If the subrun $q \xrightarrow{\lambda|w_2} q$ visits an accepting state, then we take the run $p_0 \xrightarrow{\lambda|w_1} q \xrightarrow{\lambda|w_2} q \xrightarrow{\lambda|w_4} p_n$ that is shorter and outputs no more than the original while still visiting an accepting state. If the subrun does not visit an accepting state, then we take the run $p_0 \xrightarrow{\lambda|w_1} q \xrightarrow{\lambda|w_3} q \xrightarrow{\lambda|w_4} p_n$, which is also shorter and produces no more output. Also, since we removed a subrun that does not visit an accepting state, the new run has the required property. By induction, this proves the claim.

Consider the non-deterministic real-time transducer $T' = \langle Q, A, B, \delta', q_0, F\rangle$ where $\delta'(p, a, vw, q)$ if and only if there is a state r such that $p \xrightarrow{\lambda|v} r$ with at most $2|Q|$ transitions and $r \xrightarrow{a|w} q$. From the initial claim and Lemma 4.2, it is easy to see that an infinite word x is compressed by T if and only if it is compressed by a run of T that does not use more than $2|Q|$ consecutive transitions of the form $p \xrightarrow{\lambda|v'} q$. It is also easy to check that any such run induces a run in T' which also compresses x. $\square$

Corollary 4.4. *No normal infinite word is compressible by a bounded-to-one non-real-time transducer.*

Proof. Immediate combining Theorem 4.1 and Theorem 4.3. $\square$

5. Non-deterministic counter transducers

In this section we consider non-deterministic transducers with counters. For the sake of clarity the assumption of real-time or non-real-time case is explicit.

5.1. Non-deterministic real-time counter transducers

It is straightforward to combine the proofs of incompressibility for non-deterministic real-time transducers and counter transducers, respectively Theorems 4.1 and 3.1, to obtain the following corollary.

Corollary 5.1. *No normal infinite word is compressible by a bounded-to-one non-deterministic real-time counter transducer.*

Proof. Combine the proofs of Theorem 3.1 and Theorem 4.1. The only non-trivial point to notice is that, when defining U' as in the proof of Theorem 4.1, in addition to requiring that $\langle p, \overline{m}\rangle$ be reachable, we need that there exists an infinite run starting from $\langle q, \bar{n}\rangle$ that goes infinitely often through accepting states. It is easy to check that these requirements are met and they do not invalidate any step of the proof. $\square$

5.2. Non-deterministic non-real-time counter transducers

As it stands, our proof for real-time counter transducers cannot be extended to non-real-time because in the non-real-time case the processing of a fixed word may increase or decrease the counter an unbounded amount. This voids the argument we used to give an upper bound of the set U'. On the other hand, a non-real-time transducer with two or more counters is Turing-complete [13], so it can compress computable normal infinite words. This raises the question of whether non-real-time transducers augmented with just a single counter can compress normal infinite words. We answer it in the following theorem.

A *non-real-time* 1*-counter transducer* is a tuple $T = \langle Q, A, B, \delta, q_0, F\rangle$, identical to a non-real-time transducer, but the transition δ is a subset of $Q \times \{\text{true}, \text{false}\} \times (A \cup \{\lambda\}) \times B^* \times Q \times \mathbb{Z}$. This means that the transducer can check the zero or non-zero status of the counter and modify it in each transition. The definition of compressible words is the natural one in this setting.

Theorem 5.2. *No normal infinite word is compressible by a bounded-to-one non-real-time* 1*-counter transducer.*

In the proof of Theorem 3.1 we used Lemma 3.2 to deal with counters. We bounded their increase or decrease by the length of the input word. For non-real-time transducers this argument fails. Lemma 5.7 gives an alternative argument, using Lemmas 5.3 to 5.6.

For a non-real-time 1-counter transducer T, we define D_T as for deterministic real-time counter transducers: let D_T be the maximum absolute value of a counter increment or decrement in a transition of T.

Lemma 5.3. *For every non-real-time* 1*-counter transducer T there exists another non-real-time* 1*-counter transducer T' that realizes the same relation, compresses exactly the same infinite words and has $D_{T'} \le 1$.*

Proof. We can simply emulate increasing (or decreasing) by k with k steps that do no input or output, and increase (or decrease) by 1. Since the maximum possible k is bounded by D_T, we can do it with finitely many states and transitions. Formally, if $T = \langle Q, A, B, \delta, q_0, F\rangle$ we let $T' = \langle Q \times [-D_T, D_T], A, B, \delta', \langle q_0, 0\rangle, F \times \{0\}\rangle$ where

$$\begin{aligned}\delta' = {} & \{\langle\langle q, k+1\rangle, c, \lambda, \lambda, +1, \langle q, k\rangle\rangle : 0 \le k < D_T, c \in \{\text{true}, \text{false}\}\} \;\cup \\ & \{\langle\langle q, k-1\rangle, c, \lambda, \lambda, -1, \langle q, k\rangle\rangle : D_T < k \le 0, c \in \{\text{true}, \text{false}\}\} \;\cup \\ & \{\langle\langle p, 0\rangle, c, a, v, 0, \langle q, k\rangle\rangle : \delta(p, c, a, v, k, q)\}.\end{aligned}$$

It is immediate to check that any step $\langle p, n\rangle \xrightarrow{a|v} \langle q, m\rangle$ of T induces a run $\langle\langle p, 0\rangle, n\rangle \xrightarrow{a|v} \langle\langle q, 0\rangle, m\rangle$ of at most $D_T + 1$ steps of T' and vice versa. This implies that T and T' realize the same relation and compress the same infinite words. By inspection of δ' it is clear that $D_{T'} \le 1$. □

Lemma 5.4. *Let $\langle q_0, 0\rangle \xrightarrow{a_1|v_1} \langle q_1, m_1\rangle \xrightarrow{a_2|v_2} \langle q_2, m_2\rangle \xrightarrow{a_3|v_3} \langle q_3, m_3\rangle \cdots$ be an accepting run of a bounded-to-one non-real-time 1-counter transducer. If there is a prefix $\langle q_0, 0\rangle \xrightarrow{a_1a_2\cdots a_n|v_1v_2\cdots v_n} \langle q_n, m_n\rangle$ of the run such that each accepting run that starts with it does not contain more configurations with a counter value 0, then each such accepting run does not compress $a_1a_2a_3\cdots$.*

Proof. Let $T = \langle Q, A, B, \delta, q_0, F\rangle$ be the transducer. We claim that we can emulate T on input $a_1a_2a_3\cdots$ using a transducer without a counter. Let n be as in the hypothesis and let $T' = \langle Q \cup \{r_0, r_1, \ldots, r_n\}, A, B, \delta', r_0, F\rangle$ be a non-real-time transducer, where

$$\delta' = \{p \xrightarrow{a|v} q : \exists d \in \mathbb{Z}, \delta(p, a, \text{false}, v, q, d)\} \cup \{r_i \xrightarrow{a_{i+1}|v_{i+1}} r_{i+1} : 0 \le i < n\} \cup \{r_n \xrightarrow{a_{n+1}|v_{n+1}} q_{n+1}\}$$

and $p \xrightarrow{a|v} q$ stands for the tuple $\langle p, a, v, q\rangle$. Clearly, for each accepting run of T',

$$r_0 \xrightarrow{a_1|v_1} r_1 \xrightarrow{a_2|v_2} \cdots \xrightarrow{a_n|v_n} r_n \xrightarrow{b_{n+1}|w_{n+1}} p_{n+1} \xrightarrow{b_{n+2}|w_{n+2}} \cdots$$

there is an accepting run of T,

$$\langle q_0, 0\rangle \xrightarrow{a_1|v_1} \langle q_1, m_1\rangle \xrightarrow{a_2|v_2} \cdots \xrightarrow{a_n|v_n} \langle q_n, m_n\rangle \xrightarrow{b_{n+1}|w_{n+1}} \langle p_{n+1}, m'_{n+1}\rangle \xrightarrow{b_{n+2}|w_{n+2}} \cdots$$

because by hypothesis the accepting run of T does not visit a configuration with a counter value 0 after step n, and then every transition is valid. Therefore, since T is bounded-to-one, T' is bounded-to-one. Also, consider the run of T in the hypothesis

$$\langle q_0, 0\rangle \xrightarrow{a_1a_2\cdots a_n|v_1v_2\cdots v_n} \langle q_n, m_n\rangle \xrightarrow{a_{n+1}|v_{n+1}} \langle q_{n+1}, m_{n+1}\rangle \xrightarrow{a_{n+2}|v_{n+2}} \langle q_{n+2}, m_{n+2}\rangle \cdots$$

Since $m_{n+1}, m_{n+2}, m_{n+3}, \ldots$ are all non-zero, the following run is an accepting run of T' that compresses the same input word.

$$r_0 \xrightarrow{a_1a_2\cdots a_n|v_1v_2\cdots v_n} r_n \xrightarrow{a_{n+1}|v_{n+1}} q_{n+1} \xrightarrow{a_{n+2}|v_{n+2}} q_{n+2} \cdots$$

By Corollary 4.4, T' does not compress $a_1a_2a_3\cdots$, so the given run of T does not compress it either. □

Lemma 5.5. *Let T be a non-real-time 1-counter transducer with $D_T \le 1$ and let k be a positive integer. Every finite run of T $\langle p, m\rangle \xrightarrow{u|v} \langle q, n\rangle$ such that $|m - n| \ge (k+1)(|u| + |v| + 1)$ contains a run $\langle p', m'\rangle \xrightarrow{\lambda|\lambda} \langle q', n'\rangle$ with $|m' - n'| \ge k$ and $(m - n)(m' - n') > 0$.*

Proof. From the given run $\langle p, m\rangle \xrightarrow{u|v} \langle q, n\rangle$, we extract a subrun with no input and no output. Assume $m - n \ge (k+1)(|u| + |v| + 1)$. The case $m - n \le -(k+1)(|u| + |v| + 1)$ follows by symmetry. The counter decreases by at least $(k+1)(|u| + |v| + 1)$ during the run. At most $|u| + |v|$ of that decrease happens in transitions reading some input symbol, or writing some output symbol or both. Then, the counter decreases by at least $k(|u| + |v| + 1)$ in transitions with no input nor output. Those are divided into at most $|u| + |v| + 1$ consecutive groups by the $|u| + |v|$ transitions that do input or output or both. By the pigeonhole principle, at least one of those groups decreases the counter by at least k, yielding the desired run. □

Lemma 5.6. *Let $T = \langle Q, A, B, \delta, q_0, F\rangle$ be a non-real-time 1-counter transducer with $D_T \le 1$. Let $\langle p, m\rangle \xrightarrow{u_1|v_1} \langle q_1, n_1\rangle \xrightarrow{u_2|v_2} \langle q_2, n_2\rangle \xrightarrow{u_3|v_3} \langle r, m\rangle$ be a finite run of T such that all the values of the counter are greater than $|Q|^2$ and $\min(n_1, n_2) - m \ge |Q|^2$. Then, there is another run of T, $\langle p, m\rangle \xrightarrow{u'_1|v'_1} \langle q_1, n'_1\rangle \xrightarrow{u_2|v_2} \langle q_2, n'_2\rangle \xrightarrow{u'_3|v'_3} \langle r, m\rangle$, such that $0 < n_1 - n'_1 = n_2 - n'_2 \le |Q|^2$, $|u'_1| \le |u_1|$, $|v'_1| \le |v_1|$, $|u'_3| \le |u_3|$ and $|v'_3| \le |v_3|$.*

Proof. From the run $\langle p, m\rangle \xrightarrow{u_1|v_1} \langle q_1, n_1\rangle \xrightarrow{u_2|v_2} \langle q_2, n_2\rangle \xrightarrow{u_3|v_3} \langle r, m\rangle$, we construct, as follows, a new run by removing two subruns, each of them having the same starting and ending state. This process may remove some part of the input and/or some part of the output. Let $n = \min(n_1, n_2)$. Consider the two subruns

$$\rho = \langle p, m\rangle \xrightarrow{u_1|v_1} \langle q_1, n_1\rangle \quad \text{and} \quad \sigma = \langle q_2, n_2\rangle \xrightarrow{u_3|v_3} \langle r, m\rangle.$$

Since $D_T \leq 1$, the value of the counter is set in every integer in the range $[m, n]$ in at least one configuration of ρ in increasing order. Similarly, the value of the counter is set in every integer of the range $[m, n]$ in at least one configuration of σ in decreasing order.

Let $\langle p_0, m\rangle, \langle p_1, m+1\rangle, \ldots, \langle p_{n-m}, n\rangle$ be configurations that occur in ρ in that order and let $\langle r_{n-m}, n\rangle, \langle r_{n-m-1}, n-1\rangle, \ldots, \langle r_1, m+1\rangle, \langle r_0, m\rangle$ be configurations that occur in σ in that order. Consider the pairs (p_i, r_i). There are $n-m+1$ such pairs and by hypothesis, $n-m+1 \geq |Q|^2+1$. Consider only the first $|Q|^2+1$ ones. By the pigeonhole principle, there are indices i, j with $1 \leq j-i \leq |Q|^2$ such that $p_i = p_j$ and $r_i = r_j$. Factorize the run in the hypothesis as:

$$\langle p, m\rangle \xrightarrow{u_{1,1}|v_{1,1}} \langle p_i, m+i\rangle \xrightarrow{u_{1,2}|v_{1,2}} \langle p_j, m+j\rangle \xrightarrow{u_{1,3}|v_{1,3}} \langle q_1, n_1\rangle$$
$$\xrightarrow{u_2|v_2}$$
$$\langle q_2, n_2\rangle \xrightarrow{u_{3,1}|v_{3,1}} \langle r_j, m+j\rangle \xrightarrow{u_{3,2}|v_{3,2}} \langle r_i, m+i\rangle \xrightarrow{u_{3,3}|v_{3,3}} \langle r, m\rangle.$$

Now we construct a run as required in the statement of the lemma starting from the run in the hypothesis as follows:

- subtract $j-i$ from the value of the counter to all configurations between $\langle p_j, m+j\rangle$ and $\langle r_j, m+j\rangle$ inclusive, in the original run;
- identify $\langle p_i, m+i\rangle$ with $\langle p_j, m+j-(j-i)\rangle$, and $\langle r_j, m+j-(j-i)\rangle$ with $\langle r_i, m+i\rangle$; and
- remove $\langle p_i, m+i\rangle \xrightarrow{u_{1,2}|v_{1,2}} \langle p_j, m+j-(j-i)\rangle$ and $\langle r_j, m+j-(j-i)\rangle \xrightarrow{u_{3,2}|v_{3,2}} \langle r_i, m+i\rangle$.

These modifications do not invalidate the run because $j-i \leq |Q|^2$, so, the values of the counter everywhere in the run are positive, as in the original. This yields a run that can be written using the above factorization as follows:

$$\langle p, m\rangle \xrightarrow{u_{1,1}|v_{1,1}} \langle p_i, m+i\rangle \xrightarrow{u_{1,3}|v_{1,3}} \langle q_1, n_1-j+i\rangle$$
$$\xrightarrow{u_2|v_2}$$
$$\langle q_2, n_2-j+i\rangle \xrightarrow{u_{3,1}|v_{3,1}} \langle r_i, m+i\rangle \xrightarrow{u_{3,3}|v_{3,3}} \langle r, m\rangle.$$

Let $n'_1 = n_1 - j + i$, $n'_2 = n_2 - j + i$, $u'_1 = u_{1,1}u_{1,3}$, $v'_1 = v_{1,1}v_{1,3}$, $u'_3 = u_{3,1}u_{3,3}$ and $v'_3 = v_{3,1}v_{3,3}$. This definition ensures that all the needed requirements are met. $\square$

The following lemma has the role that Lemma 3.2 had for real-time transducers, and it is the key piece in the proof of Theorem 5.2.

Lemma 5.7. *Let $T = \langle Q, A, B, \delta, q_0, F\rangle$ be a non-real-time 1-counter transducer with $D_T \leq 1$. Let $\langle q_0, 0\rangle \xrightarrow{u_1|v_1} \langle q_1, n_1\rangle \xrightarrow{u_2|v_2} \langle q_2, n_2\rangle \xrightarrow{u_3|v_3} \langle q_3, 0\rangle$ be a prefix of an accepting run of T having infinitely many configurations with a counter value 0. Then, there is a finite run of the form $\langle q_1, n'_1\rangle \xrightarrow{u_2|v_2} \langle q_2, n'_2\rangle$, with $|n'_1|, |n'_2| \leq 2(|Q|^2+1)(|u_2|+|v_2|+2)$, contained in an accepting run of T having infinitely many configurations with a counter value 0.*

Proof. Let $m = (|Q|^2+1)(|u_2|+|v_2|+2)$. Let us prove the following weaker claim: for a run

$$\rho = \langle q_0, 0\rangle \xrightarrow{u_1|v_1} \langle q_1, n_1\rangle \xrightarrow{u_2|v_2} \langle q_2, n_2\rangle \xrightarrow{u_3|v_3} \langle q_3, 0\rangle$$

with the requirements of the hypothesis, if $|n_1| > 2m$ or $|n_2| > 2m$, there is another run

$$\rho' = \langle q_0, 0\rangle \xrightarrow{u'_1|v'_1} \langle q_1, n'_1\rangle \xrightarrow{u_2|v_2} \langle q_2, n'_2\rangle \xrightarrow{u'_3|v'_3} \langle q_3, 0\rangle$$

where the same requirements hold and $|n'_1| + |n'_2| < |n_1| + |n_2|$. Then, by iterating this until $|n_1|$ and $|n_2|$ are not greater than $2m$, we obtain a run whose middle part is the run required in the statement of the lemma. We consider first $n_1 > 2m$, distinguishing two cases.

Case 1. Within $\langle q_1, n_1\rangle \xrightarrow{u_2|v_2} \langle q_2, n_2\rangle$ the counter takes a value not greater than m (possibly $n_2 = m$). Since $D_T \leq 1$, every value in the counter in the range $[m, n_1]$ occurs at least once within $\langle q_1, n_1\rangle \xrightarrow{u_2|v_2} \langle q_2, n_2\rangle$. Therefore, we can find the leftmost occurrence of the value m. Since $n_1 > 2m$, we can also find the rightmost occurrence of the value $2m$ to the left of the occurrence of the value m in the previous sentence. This yields the following factorization of $\langle q_1, n_1\rangle \xrightarrow{u_2|v_2} \langle q_2, n_2\rangle$:

$$\langle q_1, n_1\rangle \xrightarrow{u_{2,1}|v_{2,1}} \langle r_0, 2m\rangle \xrightarrow{u_{2,2}|v_{2,2}} \langle r_3, m\rangle \xrightarrow{u_{2,3}|v_{2,3}} \langle q_2, n_2\rangle$$

where $\langle r_0, 2m\rangle \xrightarrow{u_{2,2}|v_{2,2}} \langle r_3, m\rangle$ contains only configurations with values of the counter strictly between m and $2m$, with the exception of the first and last one. Apply Lemma 5.5 to the subrun $\langle r_0, 2m\rangle \xrightarrow{u_{2,2}|v_{2,2}} \langle r_3, m\rangle$ to further factorize $\langle q_1, n_1\rangle \xrightarrow{u_2|v_2} \langle q_2, n_2\rangle$ as

$$\langle q_1, n_1\rangle \xrightarrow{u_{2,1}|v_{2,1}} \langle r_0, 2m\rangle \xrightarrow{u_{2,2,1}|v_{2,2,1}} \langle r_1, k_1\rangle \xrightarrow{\lambda|\lambda} \langle r_2, k_2\rangle \xrightarrow{u_{2,2,2}|v_{2,2,2}} \langle r_3, m\rangle \xrightarrow{u_{2,3}|v_{2,3}} \langle q_2, n_2\rangle$$

with $k_1 - k_2 \geq |Q|^2 + 1$. Using $D_T \leq 1$ again, we can find the rightmost configuration of the form $\langle p, k_2\rangle$ for some state p in the run $\langle q_0, 0\rangle \xrightarrow{u_1|v_1} \langle q_1, n_1\rangle$. This shows that ρ contains a subrun

$$\begin{aligned}\rho_1 &= \langle p, k_2\rangle \xrightarrow{u_{1,2},v_{1,2}} \langle q_1, n_1\rangle \xrightarrow{u_{2,1}|v_{2,1}} \langle r_0, 2m\rangle \xrightarrow{u_{2,2,1}|v_{2,2,1}} \langle r_1, k_1\rangle \xrightarrow{\lambda|\lambda} \langle r_2, k_2\rangle \\ &= \langle p, k_2\rangle \xrightarrow{u_{1,2},v_{1,2}} \langle q_1, n_1\rangle \xrightarrow{u_{2,1}u_{2,2,1}|v_{2,1}v_{2,2,1}} \langle r_1, k_1\rangle \xrightarrow{\lambda|\lambda} \langle r_2, k_2\rangle\end{aligned}$$

with $\min(n_1, k_1) - k_2 \geq |Q|^2 + 1$ and where all the values of the counter are greater than $m \geq |Q|^2$. Therefore, we can apply Lemma 5.6 to it and obtain

$$\rho_1' = \langle p, k_2\rangle \xrightarrow{u'_{1,2},v'_{1,2}} \langle q_1, n_1'\rangle \xrightarrow{u_{2,1}u_{2,2,1}|v_{2,1}v_{2,2,1}} \langle r_1, k_1'\rangle \xrightarrow{\lambda|\lambda} \langle r_2, k_2\rangle$$

with $n_1' < n_1$. The last part of ρ_1' having no input nor output follows from the bounds on the lengths of the new input and output of the obtained run when applying Lemma 5.6.

Observe that the portions of ρ_1' that have input from u_2 or produce output to v_2 are equal to those in ρ_1. Define ρ' to be equal to ρ but replacing the subrun ρ_1 with the subrun ρ_1'. It follows that ρ' meets the required conditions.

Case 2. Within $\langle q_1, n_1\rangle \xrightarrow{u_2|v_2} \langle q_2, n_2\rangle$ the counter takes only values greater than m (including $n_2 > m$). Using $D_T \leq 1$ as before, we can get the rightmost occurrence of a configuration $\langle p, |Q|^2 + 1\rangle$ before $\langle q_1, n_1\rangle$ and the leftmost occurrence of a configuration $\langle r, |Q|^2 + 1\rangle$ after $\langle q_2, n_2\rangle$ in ρ to obtain a subrun

$$\rho_2 = \langle p, |Q|^2 + 1\rangle \xrightarrow{u_{1,2},v_{1,2}} \langle q_1, n_1\rangle \xrightarrow{u_2|v_2} \langle q_2, n_2\rangle \xrightarrow{u_{3,1}|v_{3,1}} \langle r, |Q|^2 + 1\rangle$$

where $\min(n_1, n_2) - (|Q|^2 + 1) > 2(|Q|^2 + 1) - (|Q|^2 + 1) = |Q|^2 + 1$ and all values of the counter are greater than or equal to $|Q|^2 + 1$. Therefore, we can apply Lemma 5.6 to it and obtain

$$\rho_2' = \langle p, |Q|^2 + 1\rangle \xrightarrow{u'_{1,2},v'_{1,2}} \langle q_1, n_1'\rangle \xrightarrow{u_2|v_2} \langle q_2, n_2'\rangle \xrightarrow{u'_{3,1}|v'_{3,1}} \langle r, |Q|^2 + 1\rangle$$

with $n_1' < n_1$ and $n_2' < n_2$. Define ρ' to be equal to ρ but replacing the subrun ρ_2 with the subrun ρ_2'. It follows that ρ' meets the required conditions.

The cases $n_1 < -2m$, $n_2 > 2m$ and $n_2 < -2m$ follow by symmetric arguments and using symmetric statements and proofs for the required lemmas. $\square$

We can now give the pending proof of Theorem 5.2.

Proof of Theorem 5.2. Fix a normal infinite word $x = a_1a_2\ldots$, a bounded-to-one non-real-time 1-counter transducer $T = \langle Q, A, B, \delta, q_0, F\rangle$, a real $\varepsilon > 0$ and the accepting run

$$\langle q_0, 0\rangle \xrightarrow{a_1|v_1} \langle q_1, m_1\rangle \xrightarrow{a_2|v_2} \langle q_2, m_2\rangle \xrightarrow{a_3|v_3} \langle q_3, m_3\rangle \cdots.$$

By Lemma 4.2, T reads the entire input. It suffices to show that there is a length ℓ and a set U such that Lemma 2.2 applies to this arbitrary choice of T and ε. By Lemma 5.3, let us assume without loss of generality that $D_T \leq 1$. Lemma 5.4 proves that if there is a prefix of an accepting run that cannot be extended to an accepting run visiting a configuration with a counter value 0, then the run does not compress the input. So we assume there is no such a prefix. For each word $u \in A^*$ let

$$h_u = \min\{|v| : \exists i, j, 0 \leq i \leq j, \langle q_i, m_i\rangle \xrightarrow{u|v} \langle q_j, m_j\rangle\}$$

be the minimum number of symbols that the processing of u can contribute to the output in the run we fixed. Let

$$U_\ell = \{u \in A^\ell : h_u \geq (1 - \varepsilon)\ell\}$$

be the set of words of length ℓ with relatively large contribution to the output. Let t be such that T is t-to-one. For each pair of states p, q, pair of integers c_1, c_2, word v and length ℓ, consider the set $U'(p, q, c_1, c_2, v, \ell)$ consisting of the words $u \in A^\ell$ such that there is a run $\langle p, c_1\rangle \xrightarrow{u|v} \langle q, c_2\rangle$ contained in an accepting run of T having infinitely many configurations with a counter value 0. By construction, for each $z \in U'(p, q, c_1, c_2, v, \ell)$, there is an accepting run

$$\langle q_0, 0\rangle \xrightarrow{u_0|z_0} \langle p, c_1\rangle \xrightarrow{z|v} \langle q, c_2\rangle \xrightarrow{x|y} \infty.$$

Thus, for each $z \in U'(p, q, c_1, c_2, v, \ell)$, $T(u_0zx, z_0vy)$. This implies $|U'(p, q, c_1, c_2, v, \ell)| \leq t$.

Suppose u is such that $h_u < (1 - \varepsilon)|u|$. Then, u produces an output v with $|v| < (1 - \varepsilon)\ell$ somewhere along the run we fixed; that is, there is a prefix $\langle q_0, 0\rangle \xrightarrow{u_0|v_0} \langle q_i, m_i\rangle \xrightarrow{u|v} \langle q_j, m_j\rangle$ of the fixed run. By our assumption, this prefix can be extended to a prefix of an accepting run of the form $\langle q_0, 0\rangle \xrightarrow{u_0|v_0} \langle q_i, m_i\rangle \xrightarrow{u|v} \langle q_j, m_j\rangle \xrightarrow{u_1|v_1} \langle p, 0\rangle$. Applying Lemma 5.7 to this prefix shows that $u \in U'(q_i, q_j, m_i, m_j, v, \ell)$ for $|m_i|, |m_j| \leq 2(|Q|^2 + 1)(|u| + |v| + 2)$, $|v| < (1 - \varepsilon)|u|$ and $|u| = \ell$. Thus, given u, there are at most

$$|Q|^2\big(4(|Q|^2+1)(\ell+(1-\varepsilon)\ell+2)+1\big)^2|B|^{(1-\varepsilon)\ell}$$

possible combinations for the values of q_i, q_j, m_i, m_j, v with the mentioned restrictions. Since for each such combination $|U'(q_i, q_j, m_i, m_j, v, \ell)| \le t$, there are at most

$$t|Q|^2\big(4(|Q|^2+1)(\ell+(1-\varepsilon)\ell+2)+1\big)^2|B|^{(1-\varepsilon)\ell}$$

values of w such that $h_w < (1-\varepsilon)\ell$ and $|w| = \ell$. This magnitude is $o(|A|^\ell$, so we can fix ℓ such that $|U_\ell| > |A|^\ell(1-\varepsilon)$ and take $U = U_\ell$. By construction, the fixed run of T over x fulfills the hypothesis of Lemma 2.2 using pairs of a state and a counter value as configurations. □

6. Pushdown transducers

A *pushdown transducer* is a transducer equipped with a single stack as memory that can hold elements of a finite given alphabet. The machine makes decisions based on the input and the top symbol of the stack. When moving, it can pop symbols from the stack or push symbols onto the stack. A special bottom symbol $\perp$ as top represents the empty stack. If the alphabet of the stack (symbols that can be pushed onto the stack, not including the bottom symbol) is unary, the only relevant information for the current configuration is the number of symbols contained in the stack. Also, the automaton can only test for the number to be zero (empty stack) or non-zero (non-empty stack). The stack is then equivalent to a counter with only non-negative values, although counters with positive and negative values can also be emulated with the help of the automaton states. This shows that a stack gives at least as much power as a counter.

In this section we show that both non-deterministic pushdown transducers and deterministic pushdown transducers with a single additional counter can compress normal infinite words. The latter implies that deterministic pushdown transducers with at least two stacks can compress normal infinite words. The question remains open for deterministic pushdown transducers with a single stack.

In both cases we show that a particular transducer can compress the same infinite word. Let x_0 be a normal word and let $u_i = x_0 \restriction 2^{i-1}$ be its prefix of length 2^{i-1}. We work with the infinite word $x_1 = u_1\widetilde{u}_1u_2\widetilde{u}_2u_3\widetilde{u}_3\cdots$, where $\widetilde{u}_i$ is the reverse word of u_i. It is easy to see that this infinite word is also normal, for instance with an argument similar to Champernowne's original argument [9]. The features of x_1 we exploit to provide the counterexamples are similar to the ones used by Merkle and Reimann [12]. To ease the presentation, we compress an input over alphabet A into an output over alphabet $A \cup \{\#\}$, where $\# \notin A$. However, we could have chosen any other output alphabet because there are one-to-one maps for alphabet conversion whose output/input ratio is as close to 1 as desired.

6.1. Non-deterministic pushdown transducer

In a non-deterministic pushdown transducer with a stack each transition depends on the current state, the top symbol of the stack and the input symbol. It produces an output word, a word of stack symbols that replaces the top symbol, and the new state. The transition relation δ is a finite subset of $Q \times C \times A \times B^* \times Q \times C^*$ where Q is the state set, A and B are the input and output alphabets and C is the stack alphabet. Note that the top symbol of the stack is always replaced by a word w over the stack alphabet. This means that the transducer pops the top symbol if $|w| = 0$, replaces the top symbol by another if $|w| = 1$ and pushes several symbols if $|w| > 1$.

Theorem 6.1. *There is a one-to-one non-deterministic pushdown transducer that compresses a normal infinite word.*

Proof. We give a transducer that realizes the following relation. For each input word x, the output words y satisfy $T(x, y)$ where

$$\begin{aligned} T(x,y) \Leftrightarrow\ & (x = w_1\widetilde{w}_1w_2\widetilde{w}_2\cdots w_n\widetilde{w}_n\cdots\ ,\ \ y = w_1\#w_2\#\cdots\#w_n\#\cdots) \vee \\ & (x = w_1\widetilde{w}_1w_2\widetilde{w}_2\cdots w_n\widetilde{w}_nx'\ \ ,\ \ y = w_1\#w_2\#\cdots\#w_n\#x'). \end{aligned}$$

The relation is one-to-one since x can be recovered from either $y = w_1\#w_2\#\cdots\#w_n\#\cdots$ or $y = w_1\#w_2\#\cdots\#w_n\#x'$. In the case where the input has the form $w_1\widetilde{w}_1\cdots w_n\widetilde{w}_n\cdots$, one possible output is $w_1\#\cdots\#w_n\#\cdots$. Moreover, each factor of the input $w_i\widetilde{w}_i$ produces exactly the corresponding factor of the output $w_i\#$. Let A be the input alphabet and $A \cup \{\#\}$ be the output alphabet. For i large enough, w_i is very long and $|w_i\#|\log(|A|+1)/(|w_i\widetilde{w}_i|\log|A|)$ approaches $\log(|A|+1)/(2\log|A|) < 1$. It follows that the procedure compresses the input.

The transducer proceeds as follows. It guesses non-deterministically either a factorization $x = w_1\widetilde{w}_1\cdots w_n\widetilde{w}_n\cdots$ or a factorization $x = w_1\widetilde{w}_1\cdots w_n\widetilde{w}_nx'$ of the input word x. Note that the second case is just a degenerate case of the first one where w_{n+1} becomes infinite. The transducer uses two states q_0 and q_1. It is in state q_0 when it reads a factor w_i of the remaining tail and it is in state q_1 when it reads a factor $\widetilde{w}_i$. In state q_0, each read symbol is output and pushed on the stack. The transducer non-deterministically either stays in state q_0 or moves to states q_1 to decide if the factor ended. An extra symbol # is output when it moves to state q_1. In state q_1, each read symbol is compared with the top symbol popped from the stack. If these two symbols do not match, the run fails. If they do match, nothing is output. The transducer moves back to state q_0 when the stack is empty. Note that the transducer is indeed real-time: an input symbol is read at each step of the run.

The complete transition table of the transducer is given below. The tuple $\langle p, s, a, v, q, \lambda\rangle$ is in δ if and only if the cell at row p and column $\langle s, a\rangle$ contains $\langle v, q, \text{pop}\rangle$. The tuple $\langle p, s, a, v, q, w\rangle$ is in δ for non-empty w if and only if that cell contains $\langle v, q, \text{push}\, w\rangle$. To ease the read of the table, a variable a is used in the columns, which can take the value 0 or 1, and $\bar{a}$ is $1-a$.

	$\langle a, a\rangle$	$\langle a, \bar{a}\rangle$	$\langle a, \perp\rangle$
q_0	$\langle a, q_0, \text{push}\, a\rangle$ $\langle a\#, q_1, \text{push}\, a\rangle$	$\langle a, q_0, \text{push}\, a\rangle$ $\langle a\#, q_1, \text{push}\, a\rangle$	$\langle a, q_0, \text{push}\, a\rangle$ $\langle a\#, q_1, \text{push}\, a\rangle$
q_1	$\langle \lambda, q_1, \text{pop}\rangle$		$\langle a, q_0, \text{push}\, a\rangle$ $\langle a\#, q_1, \text{push}\, a\rangle$

□

6.2. Deterministic multi-pushdown

In Theorem 3.1 we proved that real-time transducers with any number of counters cannot compress normal infinite words. Here we show that adding a single counter to a deterministic pushdown machine gives enough power to compress a normal infinite word.

A deterministic pushdown transducer with one counter uses a stack and a counter as extra memory. Each of its transitions depends on the current state, the top symbol of the stack, the zero or non-zero status of the counter, and the input symbol. It produces an output word, a word of stack symbols that replaces the top symbol, an integer to increase or decrease the counter and the new state. Thus, transitions are given by a function $\delta : Q \times C \times \{\text{true}, \text{false}\} \times A \to B^* \times Q \times C^* \times \mathbb{Z}$.

Theorem 6.2. *There is a one-to-one deterministic transducer with one stack and one counter that compresses a normal infinite word.*

Proof. We give a transducer that realizes the following function. An infinite word factorized as $w_1 w'_1 w_2 w'_2 w_3 w'_3 \cdots$ where $|w_i| = |w'_i| = 2^{i-1}$ is mapped to $w_1 v_1 \# w_2 v_2 \# w_3 v_3 \# \cdots$ where $v_i = w'_i \upharpoonright \ell_i$ is the prefix of length ℓ_i of w'_i, with ℓ_i being the length of the longest common prefix between w'_i and $\widetilde{w}_i$. Since from w_i and v_i we can easily recover w_i and w'_i, the function is one-to-one. In the case where $w'_i = \widetilde{w}_i$ the output for a prefix of the form $w_1 \widetilde{w}_1 w_2 \widetilde{w}_2 \cdots w_n \widetilde{w}_n$ is $w_1 \# w_2 \# \cdots w_n$ and therefore such an input is compressible.

The required transducer uses the counter and the stack to recognize the positions where a w_i or w'_i starts. When starting to read w_i, the counter contains $|w_i| = 2^{i-1}$ and the stack is empty. While reading each symbol of w_i we decrease the counter by 1 and push it to the stack. Therefore, the counter at 0 indicates the first symbol of w'_i and $\widetilde{w}_i$ is in the stack. While reading each symbol of w'_i, we pop from the stack and increase the counter by 2, so the empty stack indicates the beginning of w_{i+1} and the counter is left at 2^i. By default, we output λ while reading from w'_i. We compare the symbols read from w'_i and the top of the stack. When they mismatch for the first time, we output the current symbol and move to a state that does the same process, but outputs the current symbol instead of λ. When moving between states, we control the counter and the stack to avoid off-by-1 errors.

Notice that since the starting value of the counter is 0, we need a special starting state for processing w_1. The transducer has 4 states, a state q_0 to start, a state q_1 to read w_i and push, a state q_2 to read w'_i and pop while it coincides with the stack and a state q_3 to read the rest of w'_i, pop the rest of the stack and write, to use after a mismatch.

The complete transition table for the transducer is given below. The cell at row q and column $\langle a, s, c\rangle$ contains $\delta(q, s, c, a)$, where q is a state, a is a symbol from the input, s is the top symbol from the stack (or $\perp$ for empty stack) and c is represented as 0 for true and $\emptyset$ for false, to indicate zero and non-zero counter, respectively. The cells contain a tuple $\langle v, q, s, i\rangle$ where v is the word to be output, q is the new state, s is an instruction to perform in the stack as in the previous proof, and i is an increment to the counter. As before, to ease the read of the table, a variable a is used in the columns, which can take the value 0 or 1, and $\bar{a}$ is $1-a$, and an underscore (_) is also used, which means the field can take any value.

	$\langle a, \_, \_\rangle$		
q_0	$\langle a, q_1, \text{push}\, a, 0\rangle$		
	$\langle a, \_, \emptyset\rangle$	$\langle a, a, 0\rangle$	$\langle a, \bar{a}, 0\rangle$
q_1	$\langle a, q_1, \text{push}\, a, -1\rangle$	$\langle \lambda, q_2, \text{pop}, +2\rangle$	$\langle a, q_3, \text{pop}, +2\rangle$
	$\langle a, a, \_\rangle$	$\langle a, \bar{a}, \_\rangle$	$\langle a, \perp, \_\rangle$
q_2	$\langle \lambda, q_2, \text{pop}, +2\rangle$	$\langle a, q_3, \text{pop}, +2\rangle$	$\langle \# a, q_1, \text{push}\, a, -1\rangle$
	$\langle a, a, \_\rangle$	$\langle a, \bar{a}, \_\rangle$	$\langle a, \perp, \_\rangle$
q_3	$\langle a, q_3, \text{pop}, +2\rangle$	$\langle a, q_3, \text{pop}, +2\rangle$	$\langle \# a, q_1, \text{push}\, a, -1\rangle$

□

7. Selection

We consider the selection of symbols from an infinite word and define a word with the selected symbols. The general problem is which forms of selection preserve normality, that is, which families of functions f performing selection guarantee that $f(x)$ is normal when x is normal. Notice that if a selection procedure is allowed to read the symbol being decided, it would be possible to "select only zeroes", or yield similar schemes that do not preserve normality.

We consider three forms of selection. *Prefix-selection* looks at just the prefix of length $i-1$ to decide whether the symbol at position i is selected. *Suffix selection* looks at just the suffix starting at position $i+1$ to decide whether symbol at position i is selected. *Two-sided selection* looks at the prefix of length $i-1$ and the suffix starting at position $i+1$ to decide the selection of the symbol at position i. Prefix-selection generalizes the selection defined by Agafonov [1]. Suffix-selection and two-sided selection are new.

Definition. Let $x = a_1a_2a_3\cdots$ be an infinite word over alphabet A. Let $L \subseteq A^*$ be a set of finite words over A and $X \subseteq A^\omega$ a set of infinite words over A.

The word obtained by *prefix-selection* of x by L is $x \upharpoonright L = a_{p(1)}a_{p(2)}a_{p(3)}\cdots$, where $p(j)$ is the j-th smallest integer in the set $\{i : a_1a_2\cdots a_{i-1} \in L\}$.

The word obtained by *suffix-selection* of x by X is $x \upharpoonleft X = a_{p(1)}a_{p(2)}a_{p(3)}\cdots$, where $p(j)$ is the j-th smallest integer in the set $\{i : a_{i+1}a_{i+2}a_{i+3}\cdots \in X\}$.

To fix notation let us recall the definition of a finite-state automaton and the definition of a rational set of finite or infinite words.

Definition. A *finite automaton* is a tuple $S = \langle Q, A, \delta, q_0, F\rangle$ where

- Q is the set of states,
- A is the input alphabet,
- $\delta \subseteq Q \times A \times Q$ is a finite transition relation,
- $q_0 \in Q$ is the starting state and
- $F \subseteq Q$ is the set of accepting states

The automaton is deterministic if δ is a function $Q \times A \to Q$. The automaton processes symbols as the corresponding transducer, disregarding the output. The runs and accepting runs are defined as in the case of transducers.

Definition. A set of finite words L is *rational* if there is a deterministic finite automaton $S = \langle Q, A, \delta, q_0, F\rangle$ such that $L = \{u : \exists q \in F, q_0 \xrightarrow{u} q\}$. A set of infinite words X is *rational* if there is a (possibly non-deterministic) finite automaton $S = \langle Q, A, \delta, q_0, F\rangle$ such that $X = \{x : \text{there is an accepting run } q_0 \xrightarrow{x} \infty \text{ of } S\}$.

We prove Agafonov's theorem and the counterpart theorem for suffix selection. However, there are simple two-sided selections that do not preserve normality.

Theorem 7.1. *(See Agafonov [1].) Prefix selection by a rational set preserves normality.*

Theorem 7.2. *Suffix selection by a rational set preserves normality.*

Theorem 7.3. *The two-sided selection rule "select symbols in between two zeroes" does not preserve normality.*

To prove each of these three theorems we use the characterization of normality given by Theorem 1.1, which considers the limit frequency of words in a given infinite word, disregarding the positions where these words occur.

7.1. Prefix selection

To maintain the paper self-contained we include a proof of Theorem 7.1. This proof is a slight generalization of the one given by Becher and Heiber [3].

Lemma 7.4. *Let there be an accepting run of a deterministic finite automaton over a normal infinite word, such that the set of states it visits infinitely often is $\{q_1, \ldots, q_n\}$. Then, for each word u and for each of those states q_i there is another one of those states q_j such that $q_i \xrightarrow{u} q_j$.*

Proof. Let $Q_\infty = \{q_1, \ldots, q_n\}$ be the set of states visited infinitely often in the accepting run ρ. Let ρ' be a suffix of ρ such that only states in Q_∞ are visited in ρ'. Since all are visited infinitely often in ρ', it is clear that for any pair $q_i, q_j \in Q_\infty$

there is a subrun from q_i to q_j. By way of contradiction, assume the statement does not hold. Without loss of generality, assume there is u such that $q_1 \xrightarrow{u} p$ and $p \notin Q_\infty$. We build a word $u_1u_2\cdots u_n$ such that being the input to any state in Q_∞, it goes outside Q_∞. Let $u_1 = u$, so $q_1 \xrightarrow{u} p$. Inductively, consider the state p' such that $q_{i+1} \xrightarrow{u_1u_2\cdots u_i} p'$. If $p' \in Q_\infty$ then set u'_{i+1} such that $p' \xrightarrow{u'_{i+1}} q_1$ and $u_{i+1} = u'_{i+1}u$ such that $q_{i+1} \xrightarrow{u_1u_2\cdots u_iu_{i+1}} p$. If $p' \notin Q_\infty$, set $u_{i+1} = \lambda$. In both cases, we obtain $q_{i+1} \xrightarrow{u_1u_2\cdots u_iu_{i+1}} r$ with $r \notin Q_\infty$. Then, for each subrun of the form $r_1 \xrightarrow{u_1u_2\cdots u_n} r_2$, either r_1 is not in Q_∞, or $r_1 = q_i$ and $q_i \xrightarrow{u_1u_2\cdots u_i} r$ with $r \notin Q_\infty$. By normality of the input word, there are infinitely many subruns of the form $r_1 \xrightarrow{u_1u_2\cdots u_n} r_2$ in ρ'. Hence, some state not in Q_∞ is visited infinitely often in ρ', hence in ρ, contradicting the assumption. □

Lemma 7.5. *If $a_1a_2a_3\cdots$ is a normal infinite word and $q_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \xrightarrow{a_3} \cdots$ is the accepting run of a deterministic finite automaton that visits infinitely often state q_1, then,*

$$\liminf_{n\to\infty} \frac{|\{i : i \le n, p_i = q_1\}|}{n} > 0.$$

Proof. Let $Q_\infty = \{q_1, \ldots, q_n\}$ be the set of states visited infinitely often in the accepting run ρ. Let ρ' be a suffix of ρ such that only states in Q_∞ are visited in ρ'. Since all the states in Q_∞ are visited infinitely often in ρ', it is clear that for any pair $q_i, q_j \in Q_\infty$ there is a subrun from q_i to q_j. We build a word $u_1u_2\cdots u_n$ such that when it is the input to any state in Q_∞, it visits q_1. Let $u_1 = \lambda$, so $q_1 \xrightarrow{u_1} q_1$. By Lemma 7.4, $q_{i+1} \xrightarrow{u_1u_2\cdots u_i} q_j$ for some j, so let u_{i+1} be such that $q_{i+1} \xrightarrow{u_1u_2\cdots u_i} q_j \xrightarrow{u_{i+1}} q_1$. Since the automaton is deterministic, each time a subrun of the form $r_1 \xrightarrow{u_1u_2\cdots u_n} r_2$ occurs in ρ', state q_1 is visited, because if $r_1 = q_i$, the prefix $q_i \xrightarrow{u_1u_2\cdots u_i} q_1$ visits q_1 by definition. By normality of the input, $u_1u_2\cdots u_n$ occurs with a fixed positive frequency ε, so q_1 is visited at least with that same minimum frequency in ρ', and therefore in ρ. □

Lemma 7.6. *For any set of finite words L, the function $x \mapsto \langle x \upharpoonright L, x \upharpoonright A^* \setminus L\rangle$ is one-to-one.*

Proof. Let $y_1 = x \upharpoonright L$ and $y_2 = x \upharpoonright A^* \setminus L$. By definition, y_1 contains some symbols of x, in the same relative order, and y_2 contains the complement, also in the same relative order. It is possible to reconstruct x by interleaving appropriately the symbols in y_1 and y_2. For each $i \ge 1$, the i-th symbol of x comes from y_1 if and only if $x \upharpoonright (i-1) \in L$. Thus, there is a unique x such that $y_1 = x \upharpoonright L$ and $y_2 = x \upharpoonright A^* \setminus L$. □

We now introduce *two-output transducers*. These are ordinary transducers with two output tapes instead of one. In terms of compressibility they are equivalent to ordinary transducers. We give the proof for the deterministic case. It is straightforward to extend it to other cases.

Definition. A (deterministic) *two-output transducer* is a tuple $T = \langle Q, A, B, \delta, q_0\rangle$, where each element is as in deterministic transducers except the transition function $\delta : Q \times A \to B^* \times B^* \times Q$, which gives two output words.

T processes infinite words over A: if at state p symbol a is processed, T moves to state q and outputs v on tape 1 and w on tape 2, where $\langle v, w, q\rangle = \delta(p, a)$. In this case, we write $p \xrightarrow{a|v,w} q$. Finite runs, infinite runs and accepting runs are defined as for deterministic transducers. When putting together several steps, concatenation of each output is done component-wise; thus, the concatenation of the two runs

$$p_0 \xrightarrow{u_1|v_1,w_1} p_1 \quad \text{and} \quad p_1 \xrightarrow{u_2|v_2,w_2} p_2 \qquad \text{is denoted by} \qquad p_0 \xrightarrow{u_1u_2|v_1v_2,w_1w_2} p_2.$$

We write $T(x)$ to refer to the ordered pair of words such that $q_0 \xrightarrow{x|T(x)} \infty$

Definition. A two-output transducer T is *bounded-to-one* if the function $x \mapsto T(x)$ is bounded-to-one.

Definition. An infinite word $x = a_1a_2a_3\cdots$ is *compressible* by a two-output transducer if its accepting run $q_0 \xrightarrow{a_1|v_1,w_1} q_1 \xrightarrow{a_2|v_2,w_2} q_2 \xrightarrow{a_3|v_3,w_2} q_3 \cdots$ satisfies

$$\liminf_{n\to\infty} \frac{(|v_1v_2\cdots v_n| + |w_1w_2\cdots w_n|)\log|B|}{n\log|A|} < 1.$$

Theorem 7.7. *An infinite word is compressible by a bounded-to-one two-output transducer if and only if it is compressible by a bounded-to-one transducer.*

Proof. The "if" part is immediate by not using one of the output tapes. For the "only if" part, let $x = a_1a_2a_3\cdots$ be an infinite word over A compressible by a bounded-to-one two-output transducer $T = \langle Q, A, B, \delta, q_0\rangle$. We show that there is a block size m_0 such that we can interleave both outputs in blocks of m_0 symbols with one extra symbol before each block that identifies which output it came from. We maintain in the finite memory (the states) a FIFO buffer for each output tape of up to m_0 symbols. Each time we have at least m_0 symbols from the same tape, we output as many blocks as possible and drop the corresponding symbols from the buffer.

Let $b_1, b_2 \in B$ be different symbols. We use b_i to mark that a given output block comes from tape i, for $i = 1, 2$. For each positive integer m we define $O_m : B^* \times B \to B^*$ and $L_m : B^* \to B^*$, be such that, if the current buffer contains u, $O_m(u, b)$ is what is to be output, adding the extra symbol b before each block of m symbols, and $L_m(u)$ is what is left in the buffer. For each word u there is a unique factorization in $\lfloor |u|/m \rfloor$ words $u_1, u_2, \ldots, u_{\lfloor |u|/m \rfloor}$ of length m and a word v of length less than m such that $u = u_1 u_2 \cdots u_{\lfloor |u|/m \rfloor} v$. We can define O_m and L_m formally in terms of such factorization:

$$O_m(u_1 u_2 \cdots u_{\lfloor |u|/m \rfloor} v, b) = b u_1 b u_2 \cdots b u_{\lfloor |u|/m \rfloor}, \quad L_m(u_1 u_2 \cdots u_{\lfloor |u|/m \rfloor} v) = v.$$

Notice that, when $|u| < m$, $O_m(u, b) = \lambda$ and $L_m(u) = u$. Also for each positive integer m let $T_m = \langle Q \times B^{<m} \times B^{<m}, A, B, \delta_m, \langle q_0, \lambda, \lambda \rangle \rangle$ be a deterministic transducer with

$$\delta_m(\langle p, v, w \rangle, a) = \langle \langle q, L_m(vv'), L_m(ww') \rangle, O_m(vv', b_1) O_m(ww', b_2) \rangle$$

where $\delta(p, a) = \langle q, v', w' \rangle$. From an output y of T_m we can get a unique tuple of outputs y_1, y_2 of T, where y_i is the concatenation, in order, of the last m symbols of each block of $m + 1$ symbols in y that starts with b_i. Therefore, from T being bounded-to-one we can conclude each T_m is bounded-to-one. Fix an input word x and consider the accepting run of T over x,

$$q_0 \xrightarrow{a_1 | v_1, w_1} q_1 \xrightarrow{a_2 | v_2, w_2} q_2 \xrightarrow{a_3 | v_3, w_3} q_3 \cdots$$

and the accepting run of T_m over x,

$$\langle q_0, \lambda, \lambda \rangle \xrightarrow{a_1 | v'_1} C_1 \xrightarrow{a_2 | v'_2} C_2 \xrightarrow{a_3 | v'_3} C_3 \cdots$$

where each $C_i \in \{q_i\} \times B^{<m} \times B^{<m}$. By construction, the output of T_m has at most one extra symbol per m symbols of some output of T,

$$|v'_1 v'_2 \cdots v'_n| \le \frac{m+1}{m} (|v_1 v_2 \cdots v_n| + |w_1 w_2 \cdots w_n|)$$

By compressibility of x,

$$\liminf_{n \to \infty} \frac{(|v_1 v_2 \cdots v_n| + |w_1 w_2 \cdots w_n|) \log |B|}{n \log |A|} < 1.$$

Then, let m_0 be large enough such that

$$\liminf_{n \to \infty} \frac{(|v_1 v_2 \cdots v_n| + |w_1 w_2 \cdots w_n|) \log |B|}{n \log |A|} \frac{m_0 + 1}{m_0} < 1,$$

which together with the previous claim implies that T_{m_0} compresses x. $\square$

Proof of Theorem 7.1. Assume $x \in A^\omega$ is normal and $L \subseteq A^*$ is rational such that $x \upharpoonright L$ is infinite and not normal. By Lemma 7.6, $x \mapsto \langle x \upharpoonright L, x \upharpoonright A^* \setminus L \rangle$ is one-to-one. Since $x \upharpoonright L$ is not normal, there is a bounded-to-one deterministic transducer T with the same input and output alphabets that compresses it (the compressibility of non-normal words is vastly known [3]). Therefore, the function $x \mapsto \langle T(x \upharpoonright L), x \upharpoonright A^* \setminus L \rangle$ is bounded-to-one. We can compose the automaton S that accepts L with T to get a two-output transducer T' that realizes that function. It carries out S and T in parallel, sending each symbol from the input to S. If the symbol is not selected, it is output in tape 2. Else, the symbol feeds T which produces an output in tape 1.

Let $S = \langle Q_S, A, \delta_S, q_{0,S}, F \rangle$ be a deterministic automaton recognizing the rational set L and let $T = \langle Q_T, A, A, \delta_T, q_{0,T} \rangle$ be a transducer compressing $x \upharpoonright L$. The transducer T' is given by $T' = \langle Q_S \times Q_T, A, A, \delta, \langle q_{0,S}, q_{0,T} \rangle \rangle$ where

$$\begin{aligned} \delta = \{ & \langle p_s, p_t \rangle \xrightarrow{a | \lambda, a} \langle q_s, p_t \rangle : p_s \notin F, p_s \xrightarrow{a} q_s \} \cup \\ & \{ \langle p_s, p_t \rangle \xrightarrow{a | v, \lambda} \langle q_s, q_t \rangle : p_s \in F, p_s \xrightarrow{a} q_s, p_t \xrightarrow{a | v} q_t \} \end{aligned}$$

and $p \xrightarrow{a | v, w} q$ stands for the tuple $\langle p, a, v, w, q \rangle$.

Now, if $\langle q_{0,S}, q_{0,T} \rangle \xrightarrow{a_1 | v_1, w_1} \langle q_{1,S}, q_{1,T} \rangle \xrightarrow{a_2 | v_2, w_2} \langle q_{2,S}, q_{2,T} \rangle \xrightarrow{a_3 | v_3, w_3} \cdots$ is the accepting run of $x = a_1 a_2 a_3 \cdots$ on T', then by construction $q_{0,S} \xrightarrow{a_1} q_{1,S} \xrightarrow{a_2} q_{2,S} \xrightarrow{a_3} \cdots$ is a run over S. Also by construction $q_{i,S} \notin F$ implies $w_i = a_i$ and $v_i = \lambda$ and $q_{i,S} \in F$ implies $w_i = \lambda$. Then,

$$\liminf_{n \to \infty} \frac{(|v_1 v_2 \cdots v_n| + |w_1 w_2 \cdots w_n|) \log |A|}{n \log |A|} = \liminf_{n \to \infty} \frac{\sum_{i \le n : q_{i,S} \in F} |v_i| + \sum_{i \le n : q_{i,S} \notin F} 1}{n}$$

By $x \upharpoonright L$ being infinite and Lemma 7.5, there is $\varepsilon > 0$ such that

$$\liminf_{n\to\infty} |\{i \le n : q_{i,S} \in F\}|/n = \varepsilon.$$

Let $n_1, n_2, \ldots$ be an increasing sequence such that

$$\lim_{j\to\infty} |\{i \le n_j : q_{i,S} \in F\}|/n_j = \varepsilon$$

and apply that sequence to the inferior limit above, getting

$$\liminf_{n\to\infty} \frac{(|v_1 v_2 \cdots v_n| + |w_1 w_2 \cdots w_n|)\log|A|}{n \log|A|} \le \liminf_{j\to\infty} \frac{\sum_{i\le n_j : q_{i,S}\in F} |v_i| + \sum_{i \le n_j : q_{i,S}\notin F} 1}{n_j}$$

Since $\lim_{j\to\infty} \sum_{i\le n_j : q_{i,S}\notin F} 1/n_j = (1-\varepsilon)$ and $z = \lim_{j\to\infty} \frac{\sum_{i\le n_j : q_{i,S}\in F} |v_i|}{\varepsilon n_j} < 1$ because T compresses $x \upharpoonright L$, we get

$$\liminf_{n\to\infty} \frac{(|v_1 v_2 \cdots v_n| + |w_1 w_2 \cdots w_n|)\log|A|}{n \log|A|} \le \liminf_{j\to\infty} \frac{\varepsilon n_j z + (1-\varepsilon) n_j}{n_j} = \varepsilon z + (1-\varepsilon) < 1.$$

So, T' compresses x. But by Theorem 7.7 and Theorem 2.3, this is impossible. Therefore, the assumption that $x \upharpoonright L$ is not normal must be false. □

7.2. *Suffix selection*

The proof of Theorem 7.2 is similar to the one for prefix selection, but it has additional subtleties. Lemmas 7.4 and 7.5 use the determinism of the selecting automaton. This is possible because rational sets of finite words can be defined equivalently using deterministic or non-deterministic automata.

However, as we already mentioned in the Introduction, functions and relations realized by deterministic transducers are proper subclasses of rational functions and relations realized by non-deterministic ones [2]. We need slight variants of Lemmas 7.4 and 7.5 for the non-deterministic case. We will use the characterization of rational sets of infinite words that provides co-determinism instead of determinism given by Carton and Michel [8].

Definition. A *Büchi automaton* is a tuple $S = \langle Q, A, \delta, Q_0, F\rangle$ where

- Q is the set of states,
- A is the input alphabet,
- $\delta \subseteq Q \times A \times Q$ is a finite transition relation,
- $Q_0 \subseteq Q$ is the set of starting states
- $F \subseteq Q$ is the set of accepting states

The processing of the input symbols and the definition of the run coincide with those for a non-deterministic automaton. A run is accepting if it starts at *any* of the starting states and visits an accepting state infinitely often.

Note that we allow Büchi automata to have several starting states, and we use it in Theorem 7.8.

Definition. A Büchi automaton is *prophetic* if there exists exactly one run visiting a finite state infinitely often over each infinite word.

Let B be a Büchi automaton. Let X_q be the set of infinite words labeling a run starting in q and visiting a finite state infinitely often. The automaton B is prophetic if the family of sets (X_q) is a partition of the set of all infinite words.

Theorem 7.8. *(See Carton and Michel [8].) Any rational set of infinite words is accepted by a prophetic automaton.*

Proposition 7.9. *(See [8, Proposition 3].) A prophetic automaton is co-deterministic. That is, if q is a state in at least one infinite run and a is a symbol, there is exactly one state p such that $p \xrightarrow{a} q$.*

Proof. Since q is a state in at least one infinite run assume $q \xrightarrow{x} \infty$. Let $p \xrightarrow{ax} \infty$ be the only run over ax. Since it contains as a suffix a run over x, and there is only one such run, the second state in the run must be q, and thus $p \xrightarrow{a} q$. By way of contradiction, assume $p \xrightarrow{a} q$ and $p' \xrightarrow{a} q$. Then, $p \xrightarrow{ax} \infty$ and $p' \xrightarrow{ax} \infty$, contradicting the condition of the prophetic automaton. □

The next lemmas do the job of Lemmas 7.4 and 7.5 in the opposite direction. The proofs are almost the same as the ones for Lemmas 7.4 and 7.5 but using the fact that prophetic automata are co-deterministic instead of deterministic.

Lemma 7.10. *Let there be an accepting run of a prophetic automaton over a normal infinite word such that the set of states it visits infinitely often is $\{q_1,\dots,q_n\}$. Then, for each of those states q_j and each word u, there is another one of those states q_i such that $q_i \xrightarrow{u} q_j$.*

Proof. Let $Q_\infty = \{q_1,\dots,q_n\}$ be the set of states visited infinitely often in the accepting run ρ. Let ρ' be a suffix of ρ such that only states in Q_∞ are visited in ρ'. Since all are visited infinitely often in ρ', it is clear that for any pair $q_i, q_j \in Q_\infty$ there is a subrun from q_i to q_j. By way of contradiction, assume the statement does not hold, and without loss of generality, assume there is u such that $p \xrightarrow{u} q_1$ and $p \notin Q_\infty$. We build a word $u_n u_{n-1}\cdots u_1$ such that finishing its process in any state in Q_∞, it ensures a visit to a state outside Q_∞. Let $u_1 = u$, so $p \xrightarrow{u_1} q_1$. Inductively, consider the only state p' such that $p' \xrightarrow{u_i u_{i-1}\cdots u_2 u_1} q_{i+1}$. If $p' \in Q_\infty$ then set u'_{i+1} such that $q_1 \xrightarrow{u'_{i+1}} p'$ and $u_{i+1} = uu'_{i+1}$ such that $p \xrightarrow{u_{i+1}u_i\cdots u_2u_1} q_{i+1}$. If $p' \notin Q_\infty$, set $u_{i+1} = \lambda$. In both cases, we obtain $r \xrightarrow{u_{i+1}u_i\cdots u_2u_1} q_{i+1}$ with $r \notin Q_\infty$. Thus, each time there is a subrun $r_1 \xrightarrow{u_n u_{n-1}\cdots u_2 u_1} r_2$, either r_2 is not in Q_∞, or $r_2 = q_i$ and there is a prefix of the subrun $r \xrightarrow{u_i u_{i-1}\cdots u_2 u_1} r_2$ with $r \notin Q_\infty$. By normality of the input word, there are infinitely many subruns of the form $r_1 \xrightarrow{u_n u_{n-1}\cdots u_2 u_1} r_2$ in ρ'. Then, some state not in Q_∞ is visited infinitely often in ρ', and therefore in ρ, contradicting the assumption. □

Lemma 7.11. *If $a_1a_2a_3\cdots$ is a normal infinite word and $p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \xrightarrow{a_3} \cdots$ is the accepting run of a prophetic automaton that visits infinitely often state q_1, then,*

$$\liminf_{n\to\infty} \frac{|\{i : 1 \le i \le n, p_i = q_1\}|}{n} > 0.$$

Proof. Let $Q_\infty = \{q_1,\dots,q_n\}$ be the set of states visited infinitely often in the mentioned run ρ. Let ρ' be a suffix of ρ such that only states in Q_∞ are visited in ρ'. Since all are visited infinitely often in ρ', it is clear that for any pair $q_i, q_j \in Q_\infty$ there is a subrun from q_i to q_j. We build a word $u_n u_{n-1}\cdots u_1$ such that finishing its process in any state in Q_∞, it ensures a visit to q_1. Let $u_1 = \lambda$, so $q_1 \xrightarrow{u_1} q_1$. By Lemma 7.10, $q_j \xrightarrow{u_i u_{i-1}\cdots u_2 u_1} q_{i+1}$ for some j, so let u_{i+1} be such that $q_1 \xrightarrow{u_{i+1}} q_j \xrightarrow{u_i u_{i-1}\cdots u_2 u_1} q_{i+1}$. Then, since the automaton is co-deterministic, each time $r_1 \xrightarrow{u_n u_{n-1}\cdots u_2 u_1} r_2$ occurs in ρ', state q_1 is visited, because if $r_2 = q_i$, the suffix $q_1 \xrightarrow{u_i u_{i-1}\cdots u_2 u_1} r_2$ visits q_1 by definition. By normality of the input, $u_n u_{n-1}\cdots u_2 u_1$ occurs with a fixed positive frequency ε, so q_1 is visited at least with that minimum frequency on ρ', and therefore on ρ. □

When selecting with a rational set of infinite words, we need to check a prophetic automaton. Notice that the unique run for the input x contains as suffixes the unique runs for all the suffixes of x. Moreover, the run over x visits accepting states infinitely often if and only if so does each of the runs over the suffixes of x. Thus, a symbol is selected according to the state at which prophetic automaton arrives after processing it. This is the mirror of prefix selection by a finite automata, where a symbol is selected according to the state that the automaton leaves before processing it.

Two-output transducers and Theorem 7.7 can be generalized to non-deterministic transducers directly. However Lemma 7.6 cannot be used because it is based on the determinism of the automaton recognizing the rational set. A mirror Lemma 7.6 would need a finishing state, which does not exist for infinite runs. Moreover, functions of the form $x \mapsto \langle x \upharpoonright X, x \upharpoonright A^\omega \setminus X\rangle$ are not necessarily bounded-to-one, as the following example illustrates. Consider $A = \{0,1\}$ and $X = 01A^\omega$ the set of binary infinite words that start with 01. The selection rule induced by X is then the following: select symbol at position i if and only if, the two symbols at positions $i+1$ and $i+2$ are 0 and 1, respectively. Consider words $x = 000\cdots000010101\cdots$ that start with a positive number of zeroes, and then alternate zeroes and ones. It is clear that all of them get mapped by $x \mapsto \langle x \upharpoonright X, x \upharpoonright A^\omega \setminus X\rangle$ to $\langle 0111\cdots, 000\cdots\rangle = \langle 01^\omega, 0^\omega\rangle$.

To solve the last problem we insert the current state once in a while in the output in a predictable way. We also add a way to synchronize the two outputs, so that for each state we can calculate the two prefixes that were output up to that point. Then, the whole splitting process is one-to-one, because from a finishing state and two finite words we can uniquely recover the originating word by doing the same as in the proof of Lemma 7.6, but from right to left.

To add the identification for the state and the synchronization, we need to get inside the two-output merging done in the proof of Theorem 7.7, so each time we insert a block from one of the tapes, we also write the current state and the number of symbols left in the buffer of the other tape. This increases the overhead of the interleaving from 1 symbol per m symbols of input to $k + \lceil \log m \rceil$ symbols per m symbols of input, for some constant k required to encode the states. However, $(k + \lceil \log m \rceil)/m$ is also arbitrarily close to 0, so compressibility is maintained in the same way as in the proof of Theorem 7.7.

Proof of Theorem 7.2. Assume $x \in A^\omega$ is normal and X is a rational set of infinite words. By Theorem 7.8, assume X is recognized by the prophetic automaton $S = \langle Q_S, A, \delta_S, Q_{0,S}, F\rangle$. Let $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} \cdots$ be the unique run of S over $x = a_1a_2a_3\cdots$. Note that the suffix of the run $q_i \xrightarrow{a_{i+1}} q_{i+1} \xrightarrow{a_{i+2}} q_{i+2} \xrightarrow{a_{i+3}} \cdots$ is the unique run of S over the suffix of the input $a_{i+1}a_{i+2}a_{i+3}\cdots$. If finitely many of the q_i are in F, then none of these runs is accepting and $x \upharpoonright X$ is empty, and thus, finite. From now on, assume infinitely many of the q_i are in F. Symbol a_i is selected if and only if the run $q_i \xrightarrow{a_{i+1}} q_{i+1} \xrightarrow{a_{i+2}} q_{i+2} \xrightarrow{a_{i+3}} \cdots$ is accepting. Since we know it visits an accepting state infinitely often, the only additional condition is that $q_i \in Q_{0,S}$.

Assume $x \upharpoonright X$ is not normal and let $T = \langle Q_T, A, A, \delta_T, q_{0,T}\rangle$ be a bounded-to-one deterministic transducer that compresses it [3]. We build a family of bounded-to-one non-deterministic transducers T_m where each T_m simulates S and splits the output into the selected and non-selected parts x_1 and x_2, passes x_1 through T and then merges $T(x_1)$ and x_2 in blocks of m digits as in the proof of Theorem 7.7. While merging, it adds to blocks from $T(x_1)$ an indicator of the state of S where the automaton is standing, and the number of digits left in the buffer of x_2. This allows to recover, for infinitely many prefixes of each of the bounded possibilities of x_1, a corresponding prefix of x_2 and a finishing state. These, in turn, imply that the originating prefix of x is unique, making the construction bounded-to-one overall.

As in the proof of Theorem 7.7, let b_1 and b_2 be two different symbols of the output alphabet A. Let $(q_S)_{b_1,b_2}$ for a state $q_S \in Q_S$ be an injective codification of the states as integers in the range $[0, |Q_S|-1]$, written with exactly $\lceil \log |Q_S| \rceil$ binary digits. For an integer k between 0 and $m-1$, let $(k)_{b_1,b_2}$, be the binary representation of k in exactly $\lceil \log m \rceil$ binary digits. To represent binary digits over alphabet A, we write b_1 for digit 0, and b_2 for digit 1. For each positive integer m we define $O_m : A^* \times A^* \to A^*$ and $L_m : A^* \to A^*$. We use the same factorization as before, with $\lfloor |u|/m \rfloor$ words $u_1, u_2, \dots, u_{\lfloor |u|/m \rfloor}$ of length m and a word v of length less than m such that $u = u_1 u_2 \cdots u_{\lfloor |u|/m \rfloor} v$.

$$O_m(u_1 u_2 \cdots u_{\lfloor |u|/m \rfloor} v, w) = w u_1 w u_2 \cdots w u_{\lfloor |u|/m \rfloor}, \quad L_m(u_1 u_2 \cdots u_{\lfloor |u|/m \rfloor} v) = v.$$

As before, when $|u| < m$, $O_m(u, w) = \lambda$ and $L_m(u) = u$. Observe that in this case O_m can place any word before each block and not just a single symbol. Let

$$T_m = \langle Q_S \times Q_T \times A^{<m} \times A^{<m}, \delta_m, \langle q_0, q_{0,T}, \lambda, \lambda\rangle, F \times Q_T \times A^{<m} \times A^{<m}\rangle,$$

where δ_m is the set of all transitions

$$\langle p_S, p_T, v, w\rangle \xrightarrow{a|O_m(vv', b_1 (q_S)_{b_1,b_2} (|w|)_{b_1,b_2})} \langle q_S, q_T, L_m(vv'), w\rangle$$

such that $p_S \xrightarrow{a} q_S$, $q_S \in Q_{0,S}$ and $p_T \xrightarrow{a|v'} q_T$, together with the set of all transitions

$$\langle p_S, p_T, v, w\rangle \xrightarrow{a|O_m(wa, b_2)} \langle q_S, p_T, v, L_m(wa)\rangle$$

such that $p_S \xrightarrow{a} q_S$ and $q_S \notin Q_{0,S}$.

Thus, δ_m is defined as the union of two disjoint cases, the case where the current symbol is selected and the case when it is not. Since T is deterministic and the way the buffers behave encoded in L_m and O_m is also deterministic, each transition of T_m is associated in a bijective way with a transition of S. Consequently, each run of T_m is associated with a unique run of S, and thus, there is exactly one run over each input. Moreover, the unique run of T_m over x is associated to the unique run of S over x, and it has the form

$$\langle q_0, p_0, \lambda, \lambda\rangle \xrightarrow{a_1|v_1} \langle q_1, p_1, \dots\rangle \xrightarrow{a_2|v_2} \langle q_2, p_2, \dots\rangle \xrightarrow{a_3|v_3} \cdots$$

where the $\dots$ in the configurations represent the content of the buffers, which we do not record. Since infinitely many of the q_i are in F, by definition of the accepting states of T_m, infinitely many of the $\langle q_i, p_i, \dots\rangle$ in the given run are accepting. Since $\langle q_0, p_0, \lambda, \lambda\rangle$ is the starting state, the given run is an accepting run of T_m over x.

By construction, the output of T_m has an overhead of at most $k_m = 1 + \lceil \log |Q_S| \rceil + \lceil \log m \rceil$ extra symbols per m symbols of the input, while outputting exactly the input on one case (the non-selecting) and a word asymptotically shorter than the input on the other case (selecting). By Lemma 7.11 and the fact that $(m + k_m)/m$ is arbitrarily close to 1 for m large enough, it is straightforward to combine the reasoning in the proof of Theorem 7.7 and in the last part of the proof of Theorem 7.1 to show that the given run compresses the input for m large enough. Since it is an accepting run, T_m for m large enough compresses x.

To see that each T_m is bounded-to-one, assume we are given the output y and let us show that there are bounded number of possible inputs that produce it. First, parse y into blocks of $m+1$ or $m + 1 + \lceil \log |Q_S| \rceil + \lceil \log m \rceil$ symbols, where the length of each block is simply determined by its first symbol being b_1 or b_2. This uniquely reconstructs $T(x_1)$ and x_2 as mentioned above. There are a bounded number of possible x_1 because T is bounded-to-one. Fix one. Each block of $T(x_1)$ gives enough information (a pair of a state q and an integer k_1) to associate a given prefix u_1 of x_1 (the shortest prefix that produces a prefix of $T(x_1)$ to fill that many blocks of output) to a unique prefix u_2 of x_2 (if k_2 is the number of finished blocks of x_2 before u_1 is processed and k_1 is the informed integer, u_2 is exactly the first $mk_2 + k_1$ symbols of x_2) and a state of S. Since S is co-deterministic, u_1 and u_2 can be merged into a unique u, which is a prefix of the original input. This can be done for arbitrarily large prefixes, because we have the extra information infinitely often, which uniquely determines an input. In conclusion, if T is t-to-one, each T_m is also t-to-one.

We showed the existence of a bounded-to-one non-deterministic transducer that compresses a normal input x. This contradicts Theorem 4.1. Therefore, the assumption that $x \upharpoonright X$ is not normal must be false. $\square$

7.3. *Two-sided selection*

We prove here that the simple selection rule "select symbols in between two zeroes" does not preserve normality.

Proof of Theorem 7.3. Let $x = a_1a_2a_3\cdots$ be a normal infinite word over $\{0, 1\}$ and let y be the result of selecting all symbols between two zeroes, namely $y = a_{p(1)}a_{p(2)}a_{p(3)}\cdots$ where $p(j)$ is the j-th smallest integer in $\{i : a_{i-1} = a_{i+1} = 0\}$. We show that y is not normal. Let m_n be the length of the shortest prefix of x that contains n instances of 000 or 010,

$$m_n = \min\{m : |\{i : 2 \le i \le m-1, a_{i-1} = a_{i+1} = 0\}| = n\}.$$

By the normality of x, infinitely many symbols are selected. So, each m_n is well defined. Let $y = b_1b_2b_3\cdots$ and $k_n = |\{i : 1 \le i \le n-1, b_ib_{i+1} = 00\}|$. By definition of m_n and y,

$$\begin{aligned} k_n \ge\ & |\{i : 1 \le i \le m_n - 3, a_ia_{i+1}a_{i+2}a_{i+3} = 0000\}| + \\ & |\{i : 1 \le i \le m_n - 7, a_ia_{i+1}a_{i+2}a_{i+3}a_{i+4}a_{i+5}a_{i+6} = 0001000\}|. \end{aligned}$$

Therefore,

$$\begin{aligned} \lim_{n\to\infty} \frac{k_n}{n} &\ge \lim_{n\to\infty} \frac{|\{i : 1 \le i \le m_n - 3, a_ia_{i+1}a_{i+2}a_{i+3} = 0000\}|}{n} + \\ &\qquad \frac{|\{i : 1 \le i \le m_n - 7, a_ia_{i+1}a_{i+2}a_{i+3}a_{i+4}a_{i+5}a_{i+6} = 0001000\}|}{n} \\ &> \lim_{n\to\infty} \frac{|\{i : 1 \le i \le m_n - 3, a_ia_{i+1}a_{i+2}a_{i+3} = 0000\}|}{n} \\ &> \lim_{n\to\infty} \frac{|\{i : 1 \le i \le m_n - 3, a_ia_{i+1}a_{i+2}a_{i+3} = 0000\}|}{m_n} \frac{m_n}{n}. \end{aligned}$$

By definition of normality and the properties of limit,

$$\lim_{n\to\infty} \frac{|\{i : 1 \le i \le m_n - 3, a_ia_{i+1}a_{i+2}a_{i+3} = 0000\}|}{m_n} = \frac{1}{2^4} \quad \text{and} \quad \lim_{n\to\infty} \frac{m_n}{n} = 2^2.$$

This proves $\lim_{n\to\infty} k_n/n > 2^{-4}\ 2^2 = 1/4$, which implies that y is not normal. □

In the proof, we have just shown that the limit frequency of the word 00 in the output is not the expected 1/4. This suffices to prove that normality is not preserved by the selection rule. A longer and more precise calculation can show that the limit frequency is exactly 3/10. This theorem shows that preservation of normality by automata selection is limited only to prefix-selection and suffix-selection.

Acknowledgments

We thank Luc Boasson for fruitful discussions. Becher's research is supported by CONICET and Agencia Nacional de Promoción Científica y Tecnológica, Argentina. Heiber holds a fellowship from CONICET, Argentina. The three authors are members of the Laboratoire International Associé INFINIS, CONICET/Universidad de Buenos Aires–CNRS/Université Paris Diderot.

References

[1] V.N. Agafonov, Normal sequences and finite automata, Sov. Math. Dokl. 9 (1968) 324–325.
[2] M.-P. Béal, O. Carton, Determinization of transducers over infinite words: the general case, Theory Comput. Syst. 37 (4) (2004) 483–502.
[3] V. Becher, P.A. Heiber, Normal numbers and finite automata, Theor. Comput. Sci. 477 (2013) 109–116.
[4] É. Borel, Les probabilités dénombrables et leurs applications arithmétiques, Rend. Circ. Mat. Palermo 27 (1909) 247–271.
[5] A. Broglio, P. Liardet, Predictions with automata. Symbolic dynamics and its applications, Contemp. Math. 135 (1992) 111–124, Also in Proceedings AMS Conference in Honor of R.L. Adler. New Haven CT, USA, 1991.
[6] Y. Bugeaud, Distribution Modulo One and Diophantine Approximation, Cambridge Tracts in Mathematics, vol. 193, Cambridge University Press, 2012.
[7] O. Carton, P.A. Heiber, Normality and two-way automata, Inf. Comput. 241 (2015) 264–276.
[8] O. Carton, M. Michel, Unambiguous Büchi automata, Theor. Comput. Sci. 297 (2003) 37–81.
[9] D.G. Champernowne, The construction of decimals normal in the scale of ten, J. Lond. Math. Soc. 8 (1933) 254–260.
[10] J. Dai, J. Lathrop, J. Lutz, E. Mayordomo, Finite-state dimension, Theor. Comput. Sci. 310 (2004) 1–33.
[11] L. Kuipers, H. Niederreiter, Uniform Distribution of Sequences, John Wiley & Sons, New York, 1974.
[12] W. Merkle, J. Reimann, Selection functions that do not preserve normality, Theory Comput. Syst. 39 (5) (2006) 685–697.
[13] M.L. Minsky, Recursive unsolvability of Post's problem of "tag" and other topics in the theory of Turing machines, Ann. Math. 74 (1961) 437–454.
[14] M.G. O'Connor, An unpredictability approach to finite-state randomness, J. Comput. Syst. Sci. 37 (3) (1988) 324–336.
[15] Dominique Perrin, Jean-Éric Pin, Infinite Words, Elsevier, 2004.
[16] J. Sakarovitch, Elements of Automata Theory, Cambridge University Press, 2009.
[17] C.P. Schnorr, H. Stimm, Endliche Automaten und Zufallsfolgen, Acta Inform. 1 (1972) 345–359.