



DEGREES OF STREAMS**Jörg Endrullis**¹*Department of Computer Science, VU University Amsterdam, Amsterdam, The Netherlands*`j.endrullis@vu.nl`**Dimitri Hendriks**²*Department of Computer Science, VU University Amsterdam, Amsterdam, The Netherlands*`r.d.a.hendriks@vu.nl`**Jan Willem Klop***Department of Computer Science, VU University Amsterdam, Amsterdam, The Netherlands*`j.w.klop@vu.nl`

Received: 10/18/10 , Revised: 4/4/11, Accepted: 7/14/11, Published: 12/2/11

Abstract

We introduce a novel approach to comparing the complexity of streams, namely in terms of reducibility by finite state transducers. This gives rise to a hierarchy of stream ‘degrees,’ somewhat analogous to the recursion-theoretic degrees of unsolvability. It is the structure and properties of this partial order of degrees that we are primarily interested in. Remarkably, in spite of its simplicity, the main idea of this approach seems to have remained unexplored.

1. Introduction

Streams (one-sided infinite sequences of symbols) arise in several fields, ranging from formal language theory and mathematics (dynamical systems [21], fractal theory [34], number theory [36]) to physics.

Streams can be compared as to their complexity in some measure, e.g., subword complexity [15] or Kolmogorov complexity [28], but we can also consider how they can be *transformed* into each other. The latter is also the foundation of the recursion

¹Author supported by the Netherlands Organisation for Scientific Research (NWO) in project *A Term Rewriting Perspective on Program Termination*, grant number 639.021.020.

²Author supported by the Netherlands Organisation for Scientific Research (NWO) in project *Lazy Productivity* grant number 612.000.934.

theoretic degrees of unsolvability [38] that classify the intrinsic ‘difficulty’ of a set of natural numbers. Here sets of natural numbers are transformed into one another via a general algorithm (Turing machine). Another analogous hierarchy is that of Wadge degrees, an important notion in mathematical logic with implications for computational complexity, see [30]. Here subsets of spaces like the space of all 01-streams (the Cantor space) are transformed into one another using continuous functions.

We propose a novel approach to comparing stream complexity, employing finite state transducers (FSTs) to transform one stream into the other; if also a back-transformation is possible, the streams are deemed equivalent. This gives rise to a partial order, a hierarchy of degrees, that we refer to as the *FST-hierarchy*. The concept of finite automata and regular languages is a well-studied subject in formal language theory. Also finite state transducers for infinite words have gained attention. For example, in [2, Theorem 7.9.1] it has been shown that ‘morphic streams’ are closed under FSTs. For a general reference to these well-known notions, we refer to [2].

An FST is a finite automaton with output. More precisely, an FST has pairs of input letters $a \in \Sigma$ and output words $w \in \Sigma^*$ along the edges. When applied to a finite or infinite word w , the output of an FST is the concatenation of the output words encountered along the edges when reading w (in Section 5 we give the formal definition). Figure 1 depicts an FST that computes the difference of consecutive bits (the starting state is q_0). When applied to the well-known Thue–Morse sequence M , the transducer produces the period doubling sequence PD (below we state its proper definition):

$$M = 0110100110010110 \dots \qquad PD = 1011101010111011 \dots$$

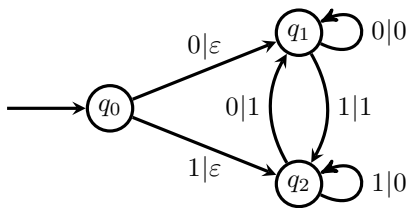


Figure 1: From Thue–Morse (M) to the period doubling sequence (PD).

We write $\sigma \triangleright \tau$ to denote that there exists an FST that transduces the stream σ to the stream τ . Then we easily have that reducibility \triangleright is reflexive and transitive, and hence we can define $\diamond = \triangleright \cap \triangleleft$ as a notion of equivalence between streams where \triangleleft is the converse of \triangleright . The equivalence classes we call *degrees*.

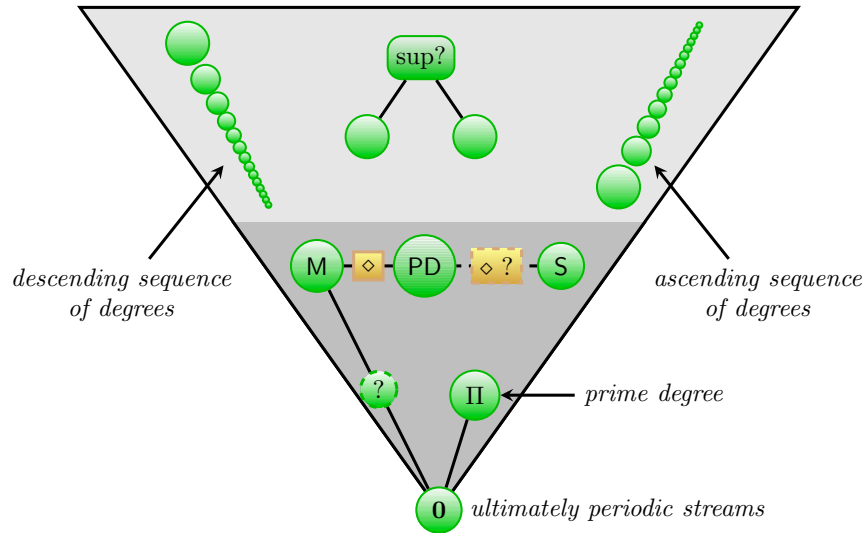


Figure 2: Uncountable partial order of stream degrees. The darker, countable part consists of morphic degrees. The Thue–Morse sequence M and the period doubling sequence PD are in the same degree. Is this a prime degree like Π ? What about the Sierpiński stream S (see Section 2)? Is S convertible with M ?

The relation \triangleright induces a partial order on the degrees. The ensuing hierarchy of degrees is sketched in Figure 2. The structure of the FST-hierarchy, as we call it, is much finer than that of the recursion-theoretic hierarchy where all computable streams are identified. For example, the Thue–Morse sequence M is no longer identified with the trivial stream of zeros $00000\dots$. We have already seen $M \triangleright PD$, and a simple exercise shows that the converse direction $PD \triangleright M$ holds as well. For this reason, we have $M \diamond PD$, that is, M belongs to the same degree as PD .

It is this partial order of degrees that we are primarily interested in. See Figure 2 which also summarizes our main results. The *bottom degree* $\mathbf{0}$ is formed by the *ultimately periodic streams*, that is, all streams σ of the form $\sigma = \tau vvv\dots$ for finite τ, v . An interesting notion that suggests itself is that of a *prime degree*: a stream σ is prime if there exists no stream τ whose degree is strictly intermediate between that of σ and the bottom degree $\mathbf{0}$. Thus the prime degrees reduce only to $\mathbf{0}$ or themselves. An example of a prime stream is:

$$\Pi = 1101001000100001000001\dots$$

Intuitively, the information content of this stream is ‘indestructible’: whatever FST is applied on Π , either the result is ultimately periodic, standing for trivial information, the $\mathbf{0}$ degree, or there is enough structure left for an FST to reconstruct

the original stream.

We briefly mention some initial results on the hierarchy (Section 6 contains the proofs):

- (i) There exist prime streams; thus the hierarchy is not dense.
- (ii) The hierarchy is not well-founded: there exist infinite descending chains.
- (iii) There are no maximal degrees: for each degree a strictly larger one can be constructed. Hence there exist infinite ascending chains.
- (iv) For every pair of streams σ, τ there exists a stream v majorizing σ and τ . However, we conjecture that not every pair of streams has a supremum as argued in Section 6.5.
- (v) Ultimate recurrence is preserved under transduction.

Let us motivate our interest in this specific hierarchy. There are various possibilities to fine-tune the recursion theoretic hierarchy. For example, one can take the (minimal) size of the programs into account for performing a certain transformation, arriving at a notion of ‘relative’ Kolmogorov complexity. Another possibility is to alter the computational model employed for transforming one stream into another. We propose the use of finite state transducers, but clearly other choices like pushdown automata would be possible as well. Our motivation for advocating the FST-hierarchy is that it has interesting properties from the perspective of infinite patterns. In a slogan one could say:

The hierarchy arising from finite state transducers classifies streams by a notion of degree that codifies their intrinsic, invariant infinite pattern.

With ‘intrinsic, invariant infinite pattern’ we mean a notion of pattern that is invariant under the insertion, removal or alteration of arbitrary finite parts of an infinite sequence. For instance, Kolmogorov complexity does not have this property; see further Section 4 for a comparison with alternative approaches. Another reason for investigating the FST-hierarchy is the exceptional simplicity and beauty of FSTs, as well as their ubiquitous presence in computer science.

Related Work

In [4] Belov studies Mealy machines and the hierarchy of streams induced by those machines. A Mealy machine (MM) is an FST with the restriction that for every input letter precisely one output letter is produced. We briefly compare the FST and MM-hierarchies.

The hierarchies have a number of properties in common, to wit: the degree $\mathbf{0}$ of ultimately periodic streams is their least element, and both hierarchies contain infinite chains and antichains.

As to their differences: For the FST-hierarchy it suffices to consider a binary alphabet (Lemma 14), whereas for the MM-hierarchy it is crucial to have alphabets of unbounded size. In the FST-hierarchy there exist prime degrees (Section 6.2), whereas in the MM-hierarchy every stream $\sigma \notin \mathbf{0}$ admits an infinite descending chain. In the MM-hierarchy suprema always exist while we conjecture that this does not hold for the FST-hierarchy.

As argued above FST-equivalence is a natural choice for a notion of stream comparison invariant under removal and insertion of finitely many elements. For instance, every stream $x = x_0x_1x_2\dots$ is FST-equivalent to its tail (shift) $\text{tail}(x) = x_1x_2\dots$. However, using Mealy machines, one can only transduce $\text{tail}(x)$ to x ; the reverse direction works only for ultimately periodic streams.

2. Defining Streams by Infinitary Term Rewriting

For defining streams and operations on them, we like to use the unifying framework of term rewriting [42]. Term rewriting not only brings a uniform notation, but also offers a framework for evaluation with a guarantee of unicity.

In previous work [18, 17] we have been interested in definability of streams by means of fixed point equations in a certain restricted format (PSF, pure stream format), restricted enough to guarantee decidability of productivity, a constructive notion of well-definedness.

Definitions of paradigm streams families abound, to wit: automatic sequences, morphic or D0L words, Toeplitz words, or streams defined by means of recurrence equations. The format PSF is expressive enough to encompass all of these classes. Also, operations realized by finite state transducers are easy to formulate in PSF.

Here we will not present the formal definition of the format, but give some examples instead. First some notation. We use $\mathbf{2}$ to denote the two-letter alphabet $\mathbf{2} = \{0, 1\}$. Let $w, v \in \mathbf{2}^*$. We write $|w|$ for the length of w . We use $w \sqsubset v$ to denote that w is a *prefix* of v , that is, there exists $u \in \mathbf{2}^*$ such that $v = wu$. We write $w|_{<n}$ to denote the *prefix of w of length n* , that is, the word $a_0\dots a_{n-1}$ where $w = a_0a_1a_2\dots$. Likewise we write $w|_{\geq n}$ for the *suffix of w starting from letter a_n* , that is, the stream $a_na_{n+1}a_{n+2}\dots$.

A stream σ is written by listing its elements:

$$\sigma(0) : \sigma(1) : \sigma(2) : \dots$$

Here the infix symbol ‘:’ is called the *stream constructor*; it prepends an element a to a stream σ to form a new stream $a : \sigma$.

<p>Thue–Morse $M = 0110100110010110\dots$ [39, A010060]</p>	<p>$M = 0 : X$ $X = 1 : \text{zip}(X, Y)$ $Y = 0 : \text{zip}(Y, X)$ $\text{zip}(x : \sigma, \tau) = x : \text{zip}(\tau, \sigma)$</p>
<p>Period doubling $PD = 0100010101000\dots$ [39, A096268]</p>	<p>$PD = \text{zip}(\text{zeros}, \text{inv}(PD))$ $\text{inv}(0 : \sigma) = 1 : \text{inv}(\sigma)$ $\text{inv}(1 : \sigma) = 0 : \text{inv}(\sigma)$ $\text{zeros} = 0 : \text{zeros}$</p>
<p>Mephisto Waltz $W = 001001110001001110\dots$ [39, A064990]</p>	<p>$W = 0 : A$ $A = 0 : 1 : \text{zip}_3(A, A, B)$ $B = 1 : 0 : \text{zip}_3(B, B, A)$ $\text{zip}_3(x : \sigma, \tau, \upsilon) = x : \text{zip}_3(\tau, \upsilon, \sigma)$</p>
<p>Kolakoski $K = 1221121221221121\dots$ [39, A000002]</p>	<p>$K = 1 : 2 : K''$ $K'' = 2 : \kappa_1(K'')$ $\kappa_1(1 : \sigma) = 1 : \kappa_2(\sigma)$ $\kappa_1(2 : \sigma) = 1 : 1 : \kappa_2(\sigma)$ $\kappa_2(1 : \sigma) = 2 : \kappa_1(\sigma)$ $\kappa_2(2 : \sigma) = 2 : 2 : \kappa_1(\sigma)$</p>
<p>Fibonacci $F = 0100101001001\dots$ [39, A003849]</p>	<p>$F = 0 : F'$ $F' = 1 : \phi(F')$ $\phi(0 : \sigma) = 0 : 1 : \phi(\sigma)$ $\phi(1 : \sigma) = 0 : \phi(\sigma)$</p>
<p>Paperfolding $PF = 0010011000110110\dots$ [39, A014707]</p>	<p>$PF = \text{zip}(\text{alt}, PF)$ $\text{alt} = 0 : 1 : \text{alt}$ $\text{zip}(x : \sigma, \tau) = x : \text{zip}(\tau, \sigma)$</p>

Table 1: Rewrite specifications of some paradigm streams. To fit in the rewrite formalism, we have to read equality ‘=’ in the table as ‘ \rightarrow ’, ‘rewrites to’.

Simple recursive equations can be used to define stream operations; for instance, the rule:

$$\text{zip}(x : \sigma, \tau) = x : \text{zip}(\tau, \sigma)$$

defines an operation $\text{zip} : A^\omega \times A^\omega \rightarrow A^\omega$ which merges two streams by taking alternately one element from σ and one from τ . This corresponds to the more

explicit definition:

$$\begin{aligned} \text{zip}(\sigma, \tau)(2n) &= \sigma(n) \\ \text{zip}(\sigma, \tau)(2n + 1) &= \tau(n) \end{aligned}$$

for all $n \in \mathbb{N}$, where $\sigma(n)$ denotes the n -th element of the stream σ .

In the same way we can define individual streams. For example:

$$M = 0 : X \qquad X = 1 : \text{zip}(X, Y) \qquad Y = 0 : \text{zip}(Y, X)$$

is an elegant specification of the Thue–Morse sequence M due to Larry Moss. We frequently use the streams `zeros` and `ones` defined by `zeros = 0 : zeros` and `ones = 1 : ones`.

Table 1 also shows PSF specifications of some other sequences: the period doubling sequence PD , the Mephisto Waltz W [25, 23], the Kolakoski sequence K , the Fibonacci word F , and the paperfolding sequence PF [11, 1, 2]. (Of course, many alternative PSF specifications exist for these paradigm streams.)

The Sierpiński Sequence

There is an intimate connection between some important fractal curves and streams via turtle graphics [22, 14]. Some sequences give rise to fractal curves by reading the terms of the sequence as drawing instructions. For example, in this way both the Thue–Morse sequence and the period doubling sequence give rise to the von Koch curve, see [31, 19]. An amusing puzzle is to derive such a sequence by looking at a curve.

In Figure 3 we have displayed the initial approximations of the Sierpiński ar-

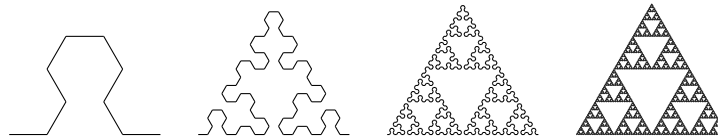


Figure 3: Construction of the Sierpiński arrowhead curve.

rowhead curve [34]. The question arises: what is the sequence behind this fractal curve? In other words, interpreting 0 and 1 as turtle drawing instructions e.g. as follows:

- 0: move forward one unit length and turn to the left $\pi/3$ radians, and
- 1: move forward one unit length and turn to the right $\pi/3$ radians,

the search is for the sequence which generates the curve, in the limit using the Hausdorff metric.

To construct the sequence, we consider Figure 3. The first iteration of the construction, the arrowhead shape, corresponds to the word $w_1 = 00111100$. The second iteration is obtained from $w_2 = w_1 0 \overline{w_1} 0 w_1 1 \overline{w_1} 1 w_1 1 \overline{w_1} 1 w_1 0 \overline{w_1} 0$ where $\overline{w_1} = 11000011$, that is, $\overline{w_1}$ is the mirrored arrowhead. Note that w_1 and $\overline{w_1}$ alternate, and the word filled in-between is w_1 itself.

The construction clearly resembles the construction of Toeplitz words, as iterated self-substitution. Toeplitz words were introduced in [24].

A Toeplitz word over an alphabet Σ is constructed as follows: Let $? \notin \Sigma$ be a new symbol, and $P \in \Sigma \cdot (\Sigma \cup \{?\})^*$, in the notation of regular expressions. Starting with the sequence $\sigma_0 = P^\omega = PPP \dots$, σ_i is obtained from σ_{i-1} by replacing the first occurrence of $?$ in σ_{i-1} by the i -th term of σ_{i-1} (which is always unequal to $?$). The *Toeplitz word generated by P* , which we denote by $T(P)$, is defined by $T(P) = \lim_{i \rightarrow \infty} \sigma_i$. Thus the sequence under construction itself is used to fill its ‘holes’, that is, replace the $?$ ’s. For example, the period doubling sequence PD is the Toeplitz word generated by the pattern $P = (010?)$:

$$P^\omega = 010?010?010?010? \dots$$

$$PD = T(P) = 0100010101000100 \dots$$

As in [1], we allow application of a letter-to-letter encoding to the read symbols that replace the $?$ ’s. For words over $\{0,1\}$, we write $\overline{?}$ to denote taking the *inverse* of the bit that is read, i.e., $\overline{0} = 1$ and $\overline{1} = 0$. In this way the pattern generating PD can be simplified to $(0\overline{?})$:

$$(0\overline{?})^\omega = 0\overline{?}0\overline{?}0\overline{?}0\overline{?}0\overline{?}0\overline{?}0\overline{?} \dots$$

$$PD = T(0\overline{?}) = 0\overline{0}\overline{0}\overline{0}\overline{0}\overline{0}\overline{0}\overline{0}\overline{0}\overline{0}\overline{0} \dots$$

$$= 0100010101000100 \dots$$

Back to the Sierpiński curve, and the sequence generating it. After close inspection, we found that the Toeplitz word generated by the pattern:

$$00111100?11000011?$$

is the desired sequence, which we call the *Sierpiński sequence S*. This pattern can be simplified to $00\overline{?}11\overline{?}$, and we define:

$$S = T(00\overline{?}11\overline{?})$$

In the rewriting format PSF the Sierpiński sequence S can be defined thus:

$$S = \text{zip}_3(\text{alt}, \text{alt}, \text{inv}(S))$$

with zip_3 , alt and inv defined by the equations in Table 1. This example shows eloquently the unifying merits of the PSF notation, as compared to the original ad hoc notation.

3. Fingerprints of Streams

Before we discuss some classical notions to compare streams in Section 4, followed by our main subject, the investigation of a new way to compare streams, namely by mutual finite state transformations, we first briefly consider a way to compare streams ‘graphically’ by displaying their orbit of iterated difference streams.

This graphical rendering lends itself to some interesting possibilities for experimentation, and we describe in this section a fortunate experiment that yielded a surprising connection between two seemingly unrelated streams. and this connection yields a useful piece of information in the FST-hierarchy introduced and analysed in Section 5.

The idea is to display in a 01-matrix $\in \mathbf{2}^{\mathbb{N} \times \mathbb{N}}$ as the first row the sequence σ that we want to ‘fingerprint’, and such that the $(n + 1)$ -th row is the difference row of the n -th. This gives a graphical impression of the ‘volatility’ of the original stream σ , at any depth. It is interesting to print the fingerprint of, e.g., the Kolakoski sequence K , totally chaotic, of the sturmian Fibonacci sequence F , looking fairly ‘quiet’ as could be expected. Let us be more precise now.

The difference operator $\delta : \mathbf{2}^\omega \rightarrow \mathbf{2}^\omega$ realized by the FST depicted in Figure 1 can be defined in the pure stream format by the rule:

$$\delta(x : y : \sigma) = (x + y) : \delta(y : \sigma)$$

for all $\sigma \in \mathbf{2}^\omega$. Here $+$ denotes addition modulo 2.

In [20] we investigated the δ -orbit $\mathcal{D}(\sigma) = (\delta^n(\sigma))_{n=0}^\infty$ of streams $\sigma \in \mathbf{2}^\omega$. We showed that the δ -orbit of an arbitrary stream σ is ultimately periodic if and only if σ is ultimately periodic. Hence the δ -orbit of M is not periodic, that is, all streams $\delta^n(M)$ are mutually different.³ A visual impression of the δ -orbit of the Morse sequence M is given by Figure 4.

Another experiment with δ -orbits is shown in Figure 5, where the δ -orbits of the Sierpiński sequence S and the Mephisto Waltz W are displayed. It is readily seen that both patterns seem identical, from the distribution of the black triangles. That they are indeed identical is revealed by a closer inspection of the first couple of rows; it turns out that the third row of the left orbit, i.e. $\delta^2(S)$, is identical to the fourth row of the right orbit, i.e. $\delta^3(W)$. Indeed, the 16×16 enlargements show at these row-positions both the prefix 1100110111100111 of length 16. We prove this observation.

Let us write σ' for the tail of a stream σ : $\sigma'(n) = \sigma(n + 1)$ for all $n \geq 0$. The zip_k operator interleaving k streams is defined by:

$$\text{zip}_k(x : \sigma_1, \sigma_2, \dots, \sigma_k) = x : \text{zip}_k(\sigma_2, \dots, \sigma_k, \sigma_1)$$

³In fact, we can give explicit expressions for the iterations of δ on M (see [26]): $\delta^{2n}(M) = \text{zip}(\delta^n(M), \delta^n(M))$ and $\delta^{2n+1}(M) = \text{zip}(\text{zeros}, \delta^n(M))$.

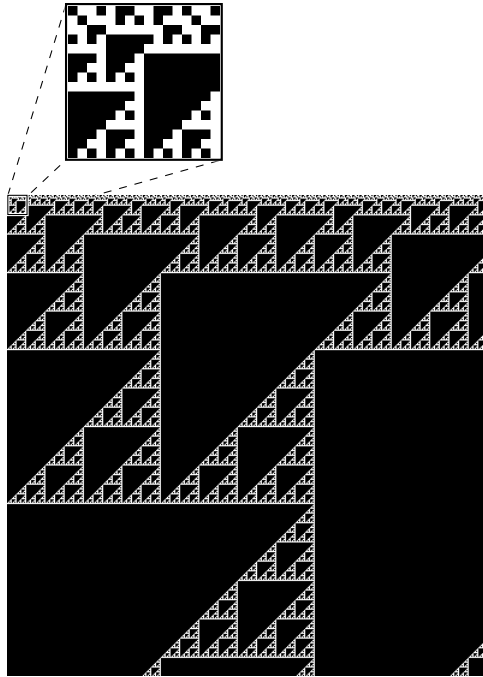


Figure 4: The first 400 iterations of δ on the Thue–Morse sequence (top row); 0s are black, 1s are white.

Moreover we let $\sigma + \tau$ denote coordinatewise addition of streams $\sigma, \tau \in \mathbf{2}^\omega$: $(\sigma + \tau)(n) = \sigma(n) + \tau(n)$. We have $\sigma + \sigma = \text{zeros}$, $\delta(\sigma) = \sigma + \sigma'$, and $\bar{\sigma} = \sigma + \text{ones}$.

Note that $\langle \mathbf{2}^\omega, +, 0^\omega \rangle$ forms a group structure, and that $(_)'$ and δ are group homomorphisms: $(\sigma + \tau)' = \sigma' + \tau'$ and $\delta(\sigma + \tau) = \delta(\sigma) + \delta(\tau)$. Moreover we have that $\text{zip}_k(\sigma_1, \dots, \sigma_k) + \text{zip}_k(\tau_1, \dots, \tau_k) = \text{zip}_k(\sigma_1 + \tau_1, \dots, \sigma_k + \tau_k)$.

We show $\delta(W) = S + \text{alt}$ from which it immediately follows that $\delta^3(W) = \delta^2(S)$, since for all $\sigma \in \mathbf{2}^\omega$ it holds that $\delta^2(\sigma + \text{alt}) = \delta^2(\sigma)$. Let $\sigma \in \mathbf{2}^\omega$. First note that $\delta(\sigma + \text{ones}) = (\sigma + \text{ones}) + (\sigma' + \text{ones}) = \sigma + \sigma' = \delta(\sigma)$. Then $\delta^2(\sigma + \text{alt}) = \delta(\sigma + \text{alt} + \sigma' + \text{alt}') = \delta(\sigma + \sigma' + \text{ones}) = \delta(\sigma + \sigma') = \delta^2(\sigma)$.

To show that $\delta(W) = S + \text{alt}$ we prove that both left and right-hand side of the equation are a solution for X in the following equation:

$$X = \text{zip}_3(\text{zeros}, \text{ones}, \overline{X})$$

which clearly has precisely one solution (namely the Toeplitz word $T(01\overline{?})$). To see this one may unfold the right-hand side to a guarded definition.

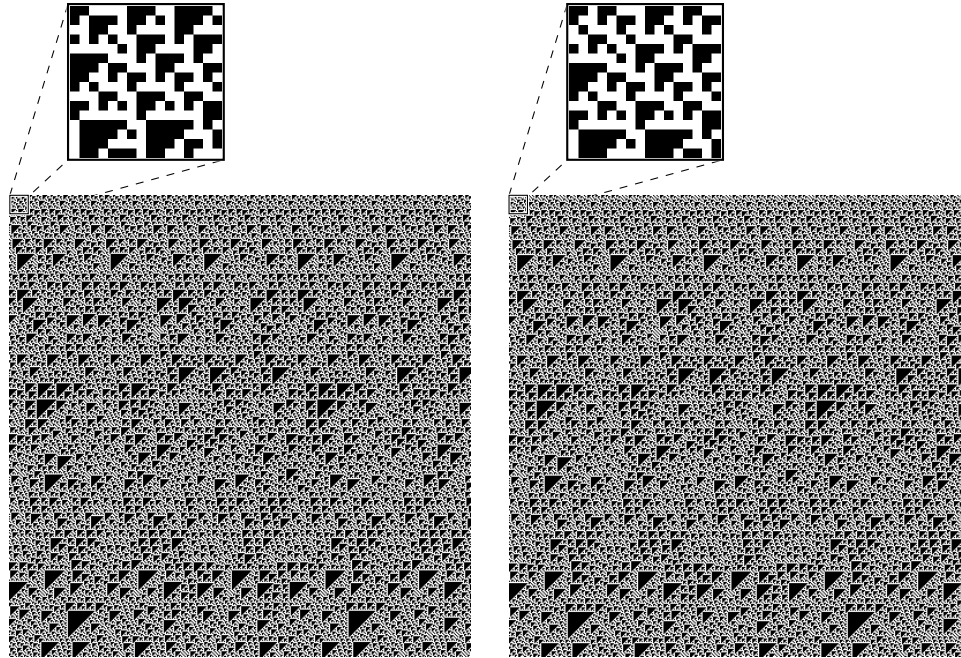


Figure 5: Comparing the ‘fingerprints’ $\mathcal{D}(S)$ and $\mathcal{D}(W)$ of the Sierpiński sequence S (left), and the Mephisto Waltz W . We find that $\delta^2(S) = \delta^3(W)$!

For W we know that $W = \text{zip}_3(W, W, \overline{W})$, and so

$$\begin{aligned} \delta(W) &= W + W' = \text{zip}_3(W, W, \overline{W}) + \text{zip}_3(W, W, \overline{W})' \\ &= \text{zip}_3(W, W, \overline{W}) + \text{zip}_3(W, \overline{W}, W') = \text{zip}_3(W + W, W + \overline{W}, \overline{W} + W') \\ &= \text{zip}_3(\text{zeros, ones, } W + W' + \text{ones}) = \text{zip}_3(\text{zeros, ones, } \overline{\delta(W)}) \end{aligned}$$

We use $S = \text{zip}_3(\text{alt, alt, } \overline{S})$ to show that also $S + \text{alt}$ is a solution for the unknown X in the equation above. We have

$$\begin{aligned} S + \text{alt} &= \text{zip}_3(\text{alt, alt, } \overline{S}) + \text{alt} = \text{zip}_3(\text{alt, alt, } \overline{S}) + \text{zip}_3(\text{alt, } \overline{\text{alt}}, \text{alt}) \\ &= \text{zip}_3(\text{alt} + \text{alt, alt} + \overline{\text{alt}}, \overline{S} + \text{alt}) = \text{zip}_3(\text{zeros, ones, } \overline{S + \text{alt}}) \end{aligned}$$

Hence W and S belong to the same degree in the FST-hierarchy.

4. Classical Complexity Notions of Streams

We briefly discuss two main notions of complexity, in order to compare these with our main subject in sections 5 and 6.

Subword Complexity

Subword complexity [15, 2, 16, 5, 36] is a classical complexity measure on infinite words σ , that records as a function of n , how many factors of σ of length n occur in σ . Thus Sturmian [29] words have subword complexity $n + 1$, automatic sequences linear, morphic words quadratic, and so on. The subword complexity of the mysterious Kolakoski stream [27, 9, 7, 33, 8, 40, 12] is unknown.

Kolmogorov Complexity

The Kolmogorov complexity [28] $\mathcal{K}(w)$ of a word w is the length of the shortest binary program computing w in a fixed universal programming system (e.g., Java, C, or a Universal Turing machine). It does not matter which system we choose, but once the choice is made it must be fixed to obtain a definite Kolmogorov complexity. All choices are also equivalent up to a constant factor.

For streams the same definition can be employed provided the streams are computable. For non-computable streams σ we can consider the function $f(n) = \mathcal{K}(\sigma|_{<n})$ where $\sigma|_{<n}$ is the prefix of σ of length n .

Kolmogorov complexity can also be employed for comparing streams, that is, computing a relative complexity. This corresponds to a fine tuning of the recursion theoretic hierarchy where one is not only interested in the existence of a finite binary program transforming one stream (equivalently, a set or function) into another, but moreover considers the size of the corresponding machines. Formally one can define the *Kolmogorov complexity* $\mathcal{K}(\sigma, \tau)$ of σ relative to τ , as the size of the smallest binary program computing σ given τ as oracle.

Comparison with the FST-hierarchy

The three notions recursion theoretic complexity, subword complexity and Kolmogorov complexity seem largely orthogonal to the complexity as in the FST-hierarchy. As mentioned in the introduction, we envisage a notion of degrees of streams classifying their infinite pattern, invariant under the exchange of finite subsequences.

The subword complexity measure is not suitable for our purposes as even non-computable streams can have linear subword complexity. To see this, take a stream σ , and define the stream τ by $\tau(2^n) = \sigma(n)$ for every $n \in \mathbb{N}$, and let $\tau(m) = 0$ for all remaining positions m . Roughly speaking, we obtain the stream τ by distributing σ sparsely over the stream of zeros. Then τ has linear subword complexity and is computable if and only if σ is computable.

Kolmogorov complexity is not appropriate as well. The reason is that the (relative) Kolmogorov complexity of a stream can be increased by an arbitrary constant by prefixing a finite word or by changing the encoding. From an infinitary point of

view, however, the streams are equivalent. For example, let M' be obtained from the Thue–Morse sequence M by applying the substitution:

$$0 \mapsto \text{I am a zero!} \qquad 1 \mapsto \text{Yes, here stands a one!}$$

Then in the (relative and non-relative) Kolmogorov complexity measure, the stream M is closer to the stream of zeros $0000\dots$ than to M' . In contrast, in the FST-hierarchy discussed below, the streams M , M' and PD are identified, and distinguished from zeros. They can easily be transformed into each other using FSTs.

5. Comparing Streams with Transducers

In this section we introduce our main subject, being the classification of streams into degrees obtained by FSTs, finite state transducers. For a thorough introduction to stream transducers, we refer to [2, 37]. We recall some of the main definitions which we employ here, for the sake of completeness, and to fix notations.

Definition 1. A (*deterministic*) *finite stream transducer (FST)* $A = \langle \Sigma, \Gamma, Q, q_0, \delta, \lambda \rangle$ consists of a finite input alphabet Σ , a finite output alphabet Γ , a finite set of states Q , an initial state $q_0 \in Q$, a transition function $\delta : Q \times \Sigma \rightarrow Q$, and an output function $\lambda : Q \times \Sigma \rightarrow \Gamma^*$.

It suffices to consider FST's with $\Sigma = \Gamma = \mathbf{2}$, see Lemma 14 below. If $\Sigma = \Gamma = \mathbf{2}$ we simply write $\langle Q, q_0, \delta, \lambda \rangle$ for the FST.

Example 2. In Figure 1 we have already seen an FST A . Formally, A can be defined as follows: $A = \langle Q, q_0, \delta, \lambda \rangle$ where $Q = \{q_0, q_1, q_2\}$, and δ and λ are given by:

$$\begin{array}{llll} \delta(q_0, 0) = q_1 & \lambda(q_0, 0) = \varepsilon & \delta(q_0, 1) = q_2 & \lambda(q_0, 1) = \varepsilon \\ \delta(q_1, 0) = q_1 & \lambda(q_1, 0) = 0 & \delta(q_1, 1) = q_2 & \lambda(q_1, 1) = 1 \\ \delta(q_2, 0) = q_1 & \lambda(q_2, 0) = 1 & \delta(q_2, 1) = q_2 & \lambda(q_2, 1) = 0 \end{array}$$

An FST A transforms a word $w = a_0a_1a_2\dots$ by reading w letter for letter. The output of A when applied to w is the concatenation $\lambda(q_0, a_0)\lambda(q_1, a_1)\dots$ of the output words encountered along the edges of A when reading w where $q_{i+1} = \delta(q_i, a_i)$ for $i = 0, 1, \dots$

Definition 3. Let $A = \langle \Sigma, \Gamma, Q, q_0, \delta, \lambda \rangle$ be an FST. We extend the state transition function δ from letters Σ to finite words Σ^* as follows:

$$\delta(q, \varepsilon) = q \qquad \delta(q, aw) = \delta(\delta(q, a), w)$$

for $q \in Q$, $a \in \Sigma$, $w \in \Sigma^*$.

The output function λ is extended to the set of all words $\Sigma^\infty = \Sigma^\omega \cup \Sigma^*$ by the following co-recursive definition:

$$\lambda(q, \varepsilon) = \varepsilon \qquad \lambda(q, aw) = \lambda(q, a) \cdot \lambda(\delta(q, a), w)$$

for $q \in Q$, $a \in \Sigma$, $w \in \Sigma^\infty$.

We introduce abbreviations $\delta(w)$ and $\lambda(w)$ as shorthand for $\delta(q_0, w)$ and $\lambda(q_0, w)$, respectively. Moreover, we define $A(\sigma) = \lambda(\sigma)$, the *output of A on $\sigma \in \Sigma^\omega$* .

Example 4. The *modified frequency modulation (MFM)* [21] is a coding scheme used for storing data on floppy disk formats and early hard discs. The MFM inserts a 0 between each two symbols unless they both are 0s, in which case it inserts a 1. This encoding guarantees that never four subsequent bits are equal, thereby allowing for easy synchronisation of the position of the read/write head. An FST implementing the MFM transformation is displayed in Figure 6.

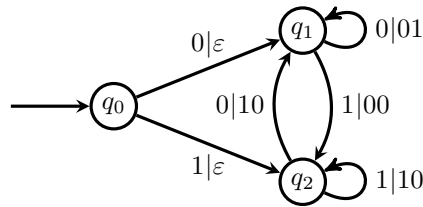


Figure 6: Modified frequency modulation.

E.g., the sequence 10100110001 is transformed into 100010010010100101001.

Remark 5. The output of an FST applied to a stream σ is not guaranteed to be a stream again. Namely, the output is a finite word if the transducer eventually visits only edges with empty output ε . For this special case, the above co-recursive definition is not ‘productive,’ and leaves $\lambda(q, \sigma)$ undefined.

For our purposes, this setup is fine as we are only interested in the hierarchy of streams (excluding finite words). Finite words would not add structure to the hierarchy as they would be situated in the bottom degree $\mathbf{0}$; FSTs can produce any finite word without even reading the given input.

Remark 6. An alternative to the co-recursive definition of $\lambda(q, \sigma)$ is to extend λ from finite words to streams σ by:

$$\lambda(q, \sigma) = \lim_{n \rightarrow \infty} \lambda(q, \sigma|_{<n}).$$

We now define ‘reducibility’ between streams in the obvious way:

Definition 7. Let $A = \langle \Sigma, \Gamma, Q, q_0, \delta, \lambda \rangle$ be an FST, $\sigma \in \Sigma^\omega$ and $\tau \in \Gamma^\omega$. We write $\sigma \triangleright_A \tau$ to denote that the A reduces σ to τ , that is, whenever $A(\sigma) = \tau$. We say σ is reducible to τ , denoted $\sigma \triangleright \tau$, if there exists an FST A such that $\sigma \triangleright_A \tau$.

The following lemma is immediate from known theory on transducers [2]:

Lemma 8. *Reducibility is reflexive and transitive, that is $\triangleright^* \subseteq \triangleright$.*

Remark 9. Transitivity of \triangleright can be shown using the ‘wreath product’ of FSTs [2]. Let $A_i = \langle \Sigma_i, \Sigma_{i+1}, Q_i, q_{i,0}, \delta_i, \lambda_i \rangle$ be FSTs for $i \in \{1, 2\}$. We define the wreath product (or composition) $A_1 \cdot A_2$ of A_1 and A_2 by:

$$\begin{aligned} A_1 \cdot A_2 &= \langle \Sigma_1, \Sigma_3, Q_1 \times Q_2, \langle q_{1,0}, q_{2,0} \rangle, \delta_{A_1 \cdot A_2}, \lambda_{A_1 \cdot A_2} \rangle \\ \delta_{A_1 \cdot A_2}(\langle q_1, q_2 \rangle, a) &= \langle \delta_1(q_1, a), \delta_2(q_2, \lambda_1(q_1, a)) \rangle \\ \lambda_{A_1 \cdot A_2}(\langle q_1, q_2 \rangle, a) &= \lambda_2(q_2, \lambda_1(q_1, a)) \end{aligned}$$

Then it is an easy exercise to check that $A_2(A_1(\sigma)) = (A_1 \cdot A_2)(\sigma)$.

Whenever we have $\sigma \triangleright \tau$ as well as a back-transformation $\tau \triangleright \sigma$, then we consider the streams σ and τ to be equivalent:

Definition 10. We define $\diamond = \triangleright \cap \triangleleft$ as the notion of *equivalence* of streams. The equivalence classes of \diamond are called *degrees*. For streams σ we use $\sigma^\diamond = \{\tau \mid \sigma \diamond \tau\}$ to denote the equivalence class of σ with respect to \diamond .

Note that \diamond is a congruence relation with respect to \triangleright . As a consequence \triangleright induces a partial order on the degrees.

Notation 11. Let $\sigma \in \mathbf{2}^\omega$ be a stream, $S, T \subseteq \mathbf{2}^\omega$ sets of streams, and $\mathbf{S}, \mathbf{T} \subseteq \mathbf{2}^{\mathbf{2}^\omega}$ sets of sets of streams (e.g., sets of degrees). We write $S \triangleright T$ if $\tau \triangleright v$ for all $\tau \in S$ and $v \in T$. We write $\mathbf{S} \triangleright \mathbf{T}$ if $S \triangleright T$ for all $S \in \mathbf{S}$ and $T \in \mathbf{T}$. Then, $\sigma \triangleright S$ is shorthand for $\{\sigma\} \triangleright S$, $S \triangleright \mathbf{T}$ for $\{S\} \triangleright \mathbf{T}$, $\sigma \triangleright \mathbf{T}$ for $\{\sigma\} \triangleright \mathbf{T}$, and likewise $S \triangleright \sigma$ for $S \triangleright \{\sigma\}$, $\mathbf{T} \triangleright S$ for $\mathbf{T} \triangleright \{S\}$, and $\mathbf{T} \triangleright \sigma$ for $\mathbf{T} \triangleright \{\sigma\}$.

We recall a few well-known definitions from partial orders:

Definition 12. Let \mathbf{S} be a set of degrees, and T a degree. The degree T is

- an *upper bound* of \mathbf{S} if $T \triangleright \mathbf{S}$,
- a *lower bound* of \mathbf{S} if $\mathbf{S} \triangleright T$,
- the *supremum* of \mathbf{S} if $T \triangleright \mathbf{S}$ and $T' \triangleright T$ for every upper bound T' of \mathbf{S} ,
- the *infimum* of \mathbf{S} if $\mathbf{S} \triangleright T$ and $T \triangleright T'$ for every lower bound T' of \mathbf{S} .

We are interested in the hierarchy of streams generated by \triangleright . When investigating the hierarchy we can, without loss of generality, restrict to streams over the alphabet $\mathbf{2}$. All other alphabets can be encoded using prefix codes:

Definition 13. A prefix code is a morphism $\phi : \Sigma \rightarrow \Gamma^*$ such that for all $a, b \in \Sigma$ with $a \neq b$ we have $\phi(a) \not\sqsupseteq \phi(b)$.

Note that for every finite alphabet Σ there exists a prefix code $\phi : \Sigma \rightarrow \mathbf{2}^*$. Then the following lemma is immediate as FSTs can convert between arbitrary prefix codes for finite alphabets:

Lemma 14. *Let Σ, Γ be finite alphabets and $\phi_\Sigma : \Sigma \rightarrow \mathbf{2}^*$, $\phi_\Gamma : \Gamma \rightarrow \mathbf{2}^*$ be prefix codes. Then we have:*

$$(\forall \sigma \in \Sigma^\omega, \tau \in \Gamma^\omega)(\sigma \triangleright \tau \iff \phi_\Sigma(\sigma) \triangleright \phi_\Gamma(\tau)).$$

Lemma 14 justifies that we restrict our investigation of the hierarchy to $\mathbf{2}^\omega$, the set of streams over $\mathbf{2}$.

Convention 15. In the sequel, we consider only stream transducers having $\mathbf{2}$ as input and output alphabet. We then write FSTs as quadruples $\langle Q, q_0, \delta, \lambda \rangle$, tacitly assuming that $\Sigma = \Gamma = \mathbf{2}$. We also restrict the hierarchy of degrees to bitstreams, then writing σ^\diamond for the set $\sigma^\diamond = \{\tau \in \mathbf{2}^\omega \mid \tau \diamond \sigma\}$.

Larger alphabets may be convenient in examples or proofs, but do not enrich the structure of the hierarchy due to the translation from larger alphabets to bitstreams given in Lemma 14.

Definition 16. Let $\mathcal{E}^\diamond = \{\sigma^\diamond \mid \sigma \in \mathbf{2}^\omega\}$ be the set of equivalence classes of \diamond .

Figure 2 on page 3 gives a pictorial impression of the FST-hierarchy \mathcal{E}^\diamond , partially ordered by \triangleright . Before we start analyzing the properties of $\langle \mathcal{E}^\diamond, \triangleright \rangle$, note how the fingerprint experiment of Section 3, yielding $\delta^3(W) = \delta^2(S)$, yields that $W \diamond S$. That is, W and S have the same degree.

6. An Initial Investigation of the Hierarchy

6.1. Immediate Observations

In this section we investigate basic properties of the FST-hierarchy. We start with a few self-evident observations.

The *bottom degree* $\mathbf{0}$ is defined to be the set of ultimately periodic streams:

$$\mathbf{0} = \{vw^\omega \mid v \in \mathbf{2}^*, w \in \mathbf{2}^+\}$$

and is the lowest degree in the FST-hierarchy:

Proposition 17. *For all degrees S we have $S \triangleright \mathbf{0}$.*

Proposition 18. *Every degree of \mathcal{E}^\diamond is countable.*

Proof. Let $\sigma \in \mathbf{2}^\omega$ be a stream. The degree σ^\diamond is countable since there are only countably many FSTs over the alphabet $\mathbf{2}$ (recall Convention 15). \square

Proposition 19. *The set \mathcal{E}^\diamond of equivalence classes is uncountable.*

Proof. Every degree is countable by Proposition 18, but there are uncountably many streams over $\mathbf{2}$. Hence \mathcal{E}^\diamond must be uncountable. \square

Proposition 20. *Every degree has only a countable set of degrees below it.*

Proof. Analogous to the proof of Proposition 18. \square

Lemma 21. *The degree $\text{zip}(\sigma, \tau)^\diamond$ is an upper bound of $\{\sigma^\diamond, \tau^\diamond\}$ all $\sigma, \tau \in \mathbf{2}^\omega$.*

The FSTs A_{even} and A_{odd} corresponding to the transformations $\text{zip}(\sigma, \tau) \triangleright \sigma$ and $\text{zip}(\sigma, \tau) \triangleright \tau$ are displayed in Figures 7 and 8, respectively.

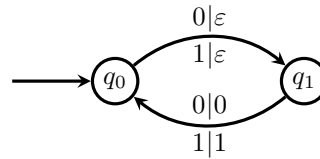
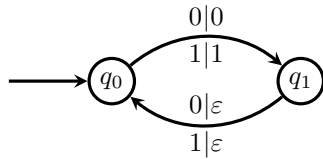


Figure 7: FST A_{even} for $\text{zip}(\sigma, \tau) \triangleright \sigma$. Figure 8: FST A_{odd} for $\text{zip}(\sigma, \tau) \triangleright \tau$.

Proposition 22. *There exist no maximal degrees.*

Proof. Let σ be a stream. The set of degrees below σ^\diamond is countable by Proposition 20, but the hierarchy of degrees is uncountable by Proposition 19. Hence there exists τ such that $\sigma \not\triangleright \tau$. Then $\text{zip}(\sigma, \tau) \triangleright \sigma$ but not $\sigma \triangleright \text{zip}(\sigma, \tau)$. \square

As a consequence we obtain:

Corollary 23. *There exist infinite ascending sequences.*

In Section 6.4 we give a constructive example of an ascending sequence.

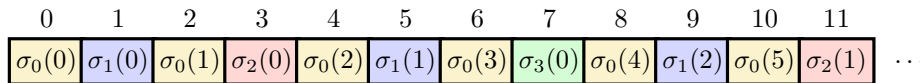


Figure 9: Zipping countably infinite families of streams: $\text{zip}((\sigma_i)_{i \in \mathbb{N}})$.

We extend the operation zip to countably infinite families of streams. Let $(\sigma_i)_{i \in \mathbb{N}}$ be a family of streams. Then we define $\text{zip}((\sigma_i)_{i \in \mathbb{N}})$ as the limit of the following

Toeplitz-like construction: Let $\tau_0 = ??? \dots$, and let τ_{i+1} be obtained from τ_i by consecutively filling the elements of σ_i into every second ?-symbol of τ_i . The initial segment of $\text{zip}((\sigma_i)_{i \in \mathbb{N}})$ is displayed in Figure 9.

Alternatively, we can define $\text{zip}((\sigma_i)_{i \in \mathbb{N}})$ using the binary zip operation:

Definition 24. Let $(\sigma_i)_{i \in \mathbb{N}}$ be a family of streams. Then we define the family of streams $(\tau_i)_{i \in \mathbb{N}}$ coinductively by $\tau_i = \text{zip}(\sigma_i, \tau_{i+1})$, and let $\text{zip}((\sigma_i)_{i \in \mathbb{N}}) = \tau_0$.

For $i \in \mathbb{N}$ the projection $\text{zip}((\sigma_i)_{i \in \mathbb{N}}) \triangleright \sigma_i$ is realized by the wreath product $A_{\text{odd}}^i \cdot A_{\text{even}}$ (with A_{even} and A_{odd} given in Figures 7 and 8). As a consequence we obtain the following lemma:

Lemma 25. *The degree $\text{zip}((\sigma_i)_{i \in \mathbb{N}})^\diamond$ is an upper bound for $\{\sigma_i^\diamond \mid i \in \mathbb{N}\}$.*

By Lemma 25, every countable set of degrees has an upper bound. The following proposition strengthens this observation by stating that the condition ‘being countable’ is not only a sufficient but also a necessary condition:

Proposition 26. *A set $S \subseteq \mathcal{E}^\diamond$ of degrees has an upper bound $\iff S$ is countable.*

Proof. The direction ‘ \Leftarrow ’ follows from Lemma 25, and ‘ \Rightarrow ’ from Proposition 20. \square

6.2. A Prime Stream

After harvesting the low hanging fruits in Section 6.1, we now turn to establishing a more challenging fact, concerned with the notion of a ‘prime degree’, that is, a degree that is minimal in the sense that it has only the bottom degree $\mathbf{0}$ below it.

Definition 27. A degree $S \in \mathcal{E}^\diamond$ is *prime* if S is not the trivial degree $\mathbf{0}$, and there is no degree between S and $\mathbf{0}$:

$$S \text{ is prime} \quad \text{if and only if} \quad S \neq \mathbf{0} \quad \text{and} \quad \neg \exists T \in \mathcal{E}^\diamond. S \triangleright T \triangleright \mathbf{0}$$

The apparent question is: *Do prime streams exist?* We give a positive answer to this question by showing that the following stream Π is prime:

$$\Pi = \prod_{k=0}^{\infty} 10^k = 1101001000100001\dots$$

That is, we prove that every result of transducing Π is either ultimately periodic or in the degree of Π itself.

The proof of primality of Π proceeds in the following steps:

- (i) We analyse the structure of reducts $\{\sigma \mid \Pi \triangleright \sigma\}$ of Π .
- (ii) For every reduct which is not ultimately periodic we prove that the breaking points of periodicity can be recognised by finite state transducers.

- (iii) The distance between the breaking points of periodicity grows linearly. Employing the fact that finite state transducers can compute inverse linear functions on the length of words, we can find a finite state transducer which reconstructs Π .

We sketch the main line of the proof. For the details, we refer to the appendix.

6.2.1. The Structure of Reducts of Π

Inspired by the structure of Π , containing blocks of zeros of increasing length, we analyse the behaviour of stream transducers on zero-words 0^* .

Definition 28. Let $A = \langle \Sigma, \Gamma, Q, q_0, \delta, \lambda \rangle$ be an FST. We define:

- A *path* of length $n \in \mathbb{N}$ in A is a sequence of pairs $\langle q_1, a_1 \rangle, \dots, \langle q_n, a_n \rangle \in Q \times \Sigma$ such that $q_{i+1} = \delta(q_i, a_i)$ for all $i = 1, \dots, n - 1$.
- A *loop* in A is a path $\langle q_1, a_1 \rangle, \dots, \langle q_n, a_n \rangle$ for which $q_1 = \delta(q_n, a_n)$, and a loop is called *minimal* if $\forall i \neq j. q_i \neq q_j$.
- A *zero-loop* in A is a minimal loop $\langle q_1, a_1 \rangle, \dots, \langle q_n, a_n \rangle$ in A that visits only edges with input 0, that is, $a_i = 0$ for all $1 \leq i \leq n$.

We use $zloops(A)$ to denote the *set of zero-loops* of A , and write Z_A for the least common multiple of the lengths of all zero-loops in $zloops(A)$.

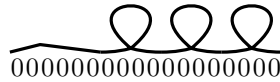


Figure 10: FST reading a word of the form 0^* .

Whenever an FST A reads a finite word $v = 000\dots$ that is longer than the number of states of A , then there must be repetition of states when A reads v . From the repetition point onwards, the automaton A is ‘caught’ in a zero-loop which A repeatedly executes, as illustrated in Figure 10. The following lemma employs this observation for the consideration of words $v0^n$ and $v0^m$ for which the difference of the lengths is a multiple of Z_A :

Lemma 29. Let $A = \langle Q, q_0, \delta, \lambda \rangle$ be an FST. For all $v \in \mathbf{2}^*$, $n \in \mathbb{N}$, $n > |Q|$ and $q \in Q$ there exist $w_1, w_2 \in \mathbf{2}^*$ such that for all $i \in \mathbb{N}$:

$$\begin{aligned} \delta(q, v 0^{n+i \cdot Z_A}) &= \delta(q, v 0^n) \\ \lambda(q, v 0^{n+i \cdot Z_A}) &= w_1 w_2^i. \end{aligned}$$

Using Lemma 29 we derive a characterisation of the form of reducts of Π , that is, streams σ for which $\Pi \triangleright \sigma$.

Proposition 30. *For every stream $\sigma \in \mathbf{2}^\omega$ with $\Pi \triangleright \sigma$ we have:*

$$\sigma = w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^n p_j \cdot c_j^i$$

for some $n \in \mathbb{N}$ and finite words $w, p_j, c_j \in \mathbf{2}^*$.

Note that $p_j \cdot c_j^i$ in Proposition 30 arises from the application of Lemma 29 to the blocks 10^* in the stream Π . To obtain the form of the double product we prove that the states of A when entering the blocks 10^* eventually start repeating periodically.

We give an example, to illustrate Proposition 30.

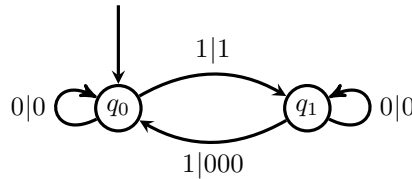


Figure 11: FST replacing every second 1 by 000.

Example 31. The FST displayed in Figure 11 replaces every second 1 by 000, that is, it reduces Π to:

$$\begin{aligned} \Pi_1 &= 100001000000001000000000001 \dots \\ &= 1 0^4 1 0^8 1 0^{12} 1 0^{16} 1 \dots \\ &= \prod_{i=0}^{\infty} (1 (00)^i \cdot 000 0 (00)^i) = \prod_{i=0}^{\infty} \prod_{j=0}^1 p_j \cdot c_j^i \end{aligned}$$

where $p_0 = 1$, $c_0 = 00$, $p_1 = 0000$ and $c_1 = 00$.

We can transform Π_1 back to Π by $0000 \mapsto 0$, a linear ‘compression’ which can easily be realised by an FST, see Lemma 35.

6.2.2. Breaking Points of Periodicity

Proposition 30 describes the general form of reducts of Π . It remains to be shown that every stream of this form that is not ultimately periodic can be transformed back to Π . For this purpose a stream transducer needs to be able to detect the transition from factors $p_j \cdot c_j^i$ to the subsequent factor $p_{j+1} \cdot c_{j+1}^i$. In general this

is not the case, as illustrated by Example 31. There it is not possible to detect the transition from $1(00)^i$ to $0000(00)^i$ since the transducers have a finite state space. However, the reduct Π_1 can be written as:

$$\Pi_1 = \prod_{i=0}^{\infty} (10000(0000)^i)$$

where each factor can be detected by the leading 1. The following proposition generalises this observation:

Proposition 32. *Let $\sigma \in \mathbf{2}^\omega$ such that σ is not ultimately periodic and:*

$$\sigma = w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^n p_j \cdot c_j^i$$

where $n \in \mathbb{N}$ and $w, p_j, c_j \in \mathbf{2}^*$ are finite words for $j = 0, \dots, n$.

Then n, w, p_j, c_j can be chosen such that:

- (i) $c_j \neq \varepsilon$ for all $j = 0, \dots, n$, and
- (ii) $p_{j+1} \not\sqsubset c_j^\omega$ for all $j = 0, \dots, n - 1$, and $p_0 \not\sqsubset c_n^\omega$.

Remark 33. As a consequence of this proposition, a finite state transducer reading σ can recognise the transition from one factor $p_j \cdot c_j^i$ to the subsequent factor $p_{j+1} \cdot c_{j+1}^i$ using bounded ‘look-ahead’. In particular, the maximum required look-ahead is $\max \{p_j \mid 0 \leq j \leq n\}$ symbols.

Although the definition of FSTs does not include look-ahead, we can simulate a look-ahead of m letters as follows. We construct an FST that reads m symbols ahead, and stores the values of the last m symbols in its memory (encoded in an enlarged state space). Then the algorithm with look-ahead can be simulated using the oldest stored letter as input, and employing the m symbols stored in memory as oracle for look-ahead.

6.2.3. Primality of Π and Density of the Hierarchy

In order to reconstruct Π , we need the following auxiliary result: FSTs can compute (inverse) linear functions on the length of words.

Definition 34. Let $f : \mathbb{N} \rightarrow \mathbb{N}$ be a function. An FST $A = \langle Q, q_0, \delta, \lambda \rangle$ is called an *f-compressor* if $\lambda(w) = 0^{f(|w|)}$ for every non-empty word $w \in \mathbf{2}^*$.

For every linear rational function f we can construct an FST A such that on the input of a word w , A produces the output $0^{f(|w|)}$.

Lemma 35. *Let $a \in \mathbb{Q}$ and $b \in \mathbb{Q}_{\geq 0}$, and define $f(n) = \max(\lfloor a + b \cdot n \rfloor, 0)$ for all $n \in \mathbb{N}$. Then there exists an f-compressor.*

We are ready for the main result:

Theorem 36. *The stream Π is prime.*

Proof. Let $\sigma \in \mathbf{2}^\omega$ such that $\Pi \triangleright \sigma$ and σ is not ultimately periodic. By Proposition 32, the stream σ can be written as:

$$\sigma = w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^n p_j \cdot c_j^i$$

where $n \in \mathbb{N}$ and $w, p_j, c_j \in \mathbf{2}^*$ for $j = 0, \dots, n$ such that $c_j \neq \varepsilon$ for all $j = 0, \dots, n$, and $p_{j+1} \not\sqsubset c_j^\omega$ for all $j = 0, \dots, n-1$, and $p_0 \not\sqsubset c_n^\omega$.

We construct a finite state automaton A that transforms σ back to Π as follows. Let A start by reading $|w|$ letters without empty output. Afterwards we let A alternating read $p_0 c_0^*, \dots, p_n c_n^*$ where A recognises the the transition from one factor $p_j c_j^*$ to $p_{j+1} c_{j+1}^*$ as described in Remark 33.

The automaton A is able to recognise the occurrences of the displayed factors $p_0 c_0^i$. Hence we obtain Π by mapping the factors $\prod_{j=0}^n p_j \cdot c_j^i$ of the outer product to 10^i for $i = 0, 1, \dots$. We have $f(i) = |\prod_{j=0}^n p_j \cdot c_j^i| = a + i \cdot b$ where $a = \sum_{j=0}^n |p_j|$ and $b = \sum_{j=0}^n |c_j|$. The inverse of f is a linear rational function: $f^{-1}(m) = (m - a)/b$. As a consequence, we obtain Π by outputting 1 at the start of each factor of the outer product, followed by 0^i constructed the f^{-1} -compressor (exists by Lemma 35) applied to the factor of the outer product. The f^{-1} -compressor can be ‘run in parallel’ with the other tasks of A by employing a cross-product like construction. \square

As a direct consequence of Theorem 36 we obtain:

Corollary 37. *The FST-hierarchy is not dense.*

6.3. An Infinite Descending Chain

We show that the hierarchy is not well-founded by proving that the following sequence of streams forms an infinite decreasing chain:

$$\begin{aligned} l_0 &= 10^{2^0} 10^{2^1} 10^{2^2} 10^{2^3} 10^{2^4} 10^{2^5} 10^{2^6} \dots \\ \triangleright l_1 &= 10^{2^0} 10^{2^2} 10^{2^4} 10^{2^6} 10^{2^8} 10^{2^{10}} 10^{2^{12}} \dots \\ \triangleright l_2 &= 10^{2^0} 10^{2^4} 10^{2^8} 10^{2^{12}} 10^{2^{16}} 10^{2^{20}} 10^{2^{24}} \dots \\ \triangleright &\dots \end{aligned}$$

where $\triangleright = (\triangleright \cap \neq)$, and for $i \in \mathbb{N}$ we define:

$$l_i = \prod_{k=0}^{\infty} 10^{(2^k \cdot 2^i)} = 10^{(2^{0 \cdot 2^i})} 10^{(2^{1 \cdot 2^i})} 10^{(2^{2 \cdot 2^i})} \dots$$

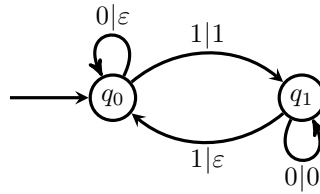


Figure 12: FST reducing l_i to l_{i+1} for all $i \in \mathbb{N}$.

Then the FST displayed in Figure 12 reduces l_i to l_{i+1} for all $i \in \mathbb{N}$. Roughly speaking, the automaton segments the streams into blocks of the form 10^* , and deletes every second block (starting from the second one).

Lemma 38. *We have $l_i \triangleright l_{i+1}$ for all $i \in \mathbb{N}$.*

Proof. Let $i \in \mathbb{N}$; we apply the automaton A from Figure 12 to l_i . As discussed above, the A drops every second block of the form 10^* . Hence we obtain

$$l_i = \prod_{k=0}^{\infty} 10^{(2^k \cdot 2^i)} \triangleright_A \prod_{k=0}^{\infty} 10^{(2^{2^k \cdot 2^i})} = \prod_{k=0}^{\infty} 10^{(2^k \cdot 2^{i+1})} = l_{i+1}.$$

□

Hence $(l_i)_{i \in \mathbb{N}}$ forms an infinite \triangleright chain. It remains to be shown that each of the \triangleright -steps is strict, that is, a \triangleright -step.

Lemma 39. *Let $k \in \mathbb{N}$. For every stream $\sigma \in \mathbf{2}^\omega$ with $l_k \triangleright \sigma$ we have:*

$$\sigma = w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^n p_j \cdot c_j^{f(i \cdot (n+1) + j)}$$

for some $n \in \mathbb{N}$ and words $w, p_j, c_j \in \mathbf{2}^*$, and $f : \mathbb{N} \rightarrow \mathbb{N}$ with $f(m) \in \Theta(2^{m \cdot 2^k})$.

Proof. The proof is analogous to the proof of Lemma 30. Let $k \in \mathbb{N}$, $\sigma \in \mathbf{2}^\omega$, and $A = \langle Q, q_0, \delta, \lambda \rangle$ be an FST such that $l_k \triangleright_A \sigma$. We have:

$$l_k = 10^{(2^0 \cdot 2^k)} 10^{(2^1 \cdot 2^k)} 10^{(2^2 \cdot 2^k)} \dots$$

We consider l_k as sequence of blocks $\gamma_i = 10^{(2^i \cdot 2^k)}$. Let q_i denote that state of A when entering the block γ_i (during reading σ). By the Pigeonhole Principle there exist $n_1, n_2 \in \mathbb{N}$ such that $|Q| < 2^{n_1} < 2^{n_2}$, and $2^{n_1} \equiv 2^{n_2} \pmod{Z_A}$, and $q_{n_1} = q_{n_2}$. Then $2^{n_1+i} \equiv 2^{n_2+i} \pmod{Z_A}$ for all $i \in \mathbb{N}$. Hence $q_{n_1+i} = q_{n_2+i}$ for all $i \in \mathbb{N}$ as a consequence of Lemma 29.

By Lemma 29 there exist $p_j, c_j \in \mathbf{2}^*$ for $j = 0, \dots, n_2 - n_1$ such that: whenever $m = n_1 + i \cdot (n_2 - n_1) + j$ with $i \in \mathbb{N}$ and $0 \leq j < n_2 - n_1$, then

$$\lambda(q_m, \gamma_m) = \lambda(q_{n_i+j}, \gamma_m) = \lambda(q_{n_i+j}, \gamma_{n_1+j} 0^{\ell \cdot Z_A}) = p_j \cdot c_j^\ell$$

where $\ell \in \mathbb{N}$ such that $2^{m \cdot 2^k} - 2^{(n_1+j) \cdot 2^k} = \ell \cdot Z_A$. Thereby we have implicitly defined the mapping $f : \mathbb{N} \rightarrow \mathbb{N}$ by $(m - n_1) \mapsto \ell$. Note that $f \in \Theta(2^{(m-n_1) \cdot 2^k})$, which yields the claim. \square

Theorem 40. $l_0 \triangleright l_1 \triangleright l_2 \triangleright \dots$ is an infinite decreasing chain.

Proof. We have $l_i \triangleright l_{i+1}$ for all $i \in \mathbb{N}$ by Lemma 38. Assume there exists $k \in \mathbb{N}$ such that $l_{k+1} \triangleright l_k$. Then by Lemma 39 l'_k must be of the form

$$l'_k = w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^n p_j \cdot c_j^{f(i \cdot (n+1) + j)}$$

with $f(m) \in \Theta(2^{m \cdot 2^{k+1}})$.

Since for every $\ell \in \mathbb{N}$, $1 0^\ell 1$ occurs at most once in l_k , it follows that for all $j = 0, \dots, n$, p_j contains at most one 1, and c_j consists only of zeros. However, there must exist one $j \in \{0, \dots, n\}$ such that $c_j \neq \varepsilon$. Hence, the displayed occurrences of $\gamma_i = c_j^{f(i \cdot (n+1) + j)}$ for $i = 0, 1, 2, \dots$ are separated by at most $n + 1$ ones, but the number of zeros in γ_i grows with speed $\Theta((2^{2^{k+1} \cdot (n+1)})^i)$. This is faster than the growth of the corresponding blocks in l_k : $\Theta((2^{2^k \cdot (n'+1)})^i)$ with $n' \leq n$. This contradicts the assumption $l'_k = l_k$. \square

6.4. An Infinite Ascending Chain

In this section, we construct an infinite ascending chain. The existence of such a chain was non-constructively proven in Corollary 23. The family $(U_k)_{k \in \mathbb{N}}$ establishes a concrete example of an infinite ascending chain:

$$\begin{aligned} & \dots \\ \triangleright U_2 &= 1(10)^2 1(100)^2 1(10000)^2 1(100000000)^2 \dots \\ \triangleright U_1 &= 110 1100 110000 1100000000 \dots \\ \triangleright U_0 &= 111111 \dots \end{aligned}$$

where for $k \in \mathbb{N}$ we define:

$$U_k = \prod_{i=0}^{\infty} (1 \cdot \prod_{j=0}^{k-1} 10^{(2^j)}) = 1(10)^k 1(100)^k 1(10000)^k 1(100000000)^k \dots$$

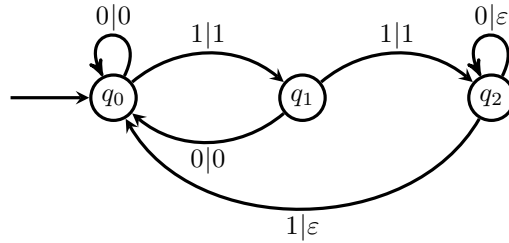


Figure 13: FST reducing U_{k+1} to U_k for all $k \in \mathbb{N}$.

We can transform U_{k+1} to U_k by mapping $110^*1 \mapsto 11$, that is, we remove the first block 10^* after every occurrence of 11 . This transformation is implemented by the FST displayed in Figure 13.

Then we immediately obtain the following lemma:

Lemma 41. *We have $U_{k+1} \triangleright U_k$ for all $k \in \mathbb{N}$.*

Theorem 42. *$U_0 \triangleleft U_1 \triangleleft U_2 \triangleleft \dots$ is an infinite ascending chain.*

Proof. We have $U_{k+1} \triangleright U_k$ for all $k \in \mathbb{N}$ by Lemma 41. Assume that there exists $k \in \mathbb{N}$ and an FST $A = \langle Q, q_0, \delta, \lambda \rangle$ such that $U_k \triangleright_A U_{k+1}$. Then $k > 0$ since U_0 is ultimately periodic while U_1 is not. We consider U_k as a sequence of blocks γ_i of the form 110^+ and 10^+ . Let q_i be the state of A when entering γ_i .

By the Pigeon Hole Principle there exist $n_1, n_2 \in \mathbb{N}$ such that $|Q| < 2^{n_1} < 2^{n_2}$, and $2^{n_1} \equiv 2^{n_2} \pmod{Z_A}$, and A enters the block $110^{2^{n_1}}$ in the same state as $110^{2^{n_2}}$. Let m_1 be the index of block $110^{2^{n_1}}$ and m_2 the index of block $110^{2^{n_2}}$, that is, $\gamma_{m_1} = 110^{2^{n_1}}$ and $\gamma_{m_2} = 110^{2^{n_2}}$. Define $p = m_2 - m_1$. Then $q_{m+p} = q_m$ for every $m \geq m_1$. Then:

$$\lambda(U_k) = w \cdot \lambda(q_{m_1}, \gamma_{m_1}) \cdot \lambda(q_{m_1+1}, \gamma_{m_1+1}) \dots$$

By Lemma 29 for every $q \in Q$ there exist p_q, c_q, p'_q, c'_q such that:

$$\begin{aligned} \lambda(q, 10^{|Q|+\ell \cdot Z_A}) &= p_q \cdot c_q^\ell \\ \lambda(q, 110^{|Q|+\ell \cdot Z_A}) &= p'_q \cdot c'_q{}^\ell \end{aligned}$$

As in the proof of Theorem 40 it follows that for every $q \in Q$, p_q contains at most one occurrence of 11 or 1 , and c_q consists only of zeros if $\lambda(q, 10^*)$ occurs in the periodic part, and likewise for p'_q , and c'_q if $\lambda(q, 110^*)$ occurs in the periodic part. As a consequence, from γ_{m_1} onwards, one block of U_k gets mapped to at most one block of $\lambda(U_k)$ (no fresh delimiters 11 or 1 are created).

Again, since U_{k+1} is not ultimately periodic, there exists $q \in Q$ such that either $c_q \neq \varepsilon$ and $\lambda(q, 10^*)$ occurs in the periodic part, or $c'_q \neq \varepsilon$ and $\lambda(q, 110^*)$ occurs

in the periodic part. Without loss of generality assume $c_q \neq \varepsilon$. Then we have $|\lambda(q, 10^m)| \in \Theta(m)$. As in the proof of Theorem 40 this leads to a contradiction as the size of the blocks in $\lambda(U_k)$ grows faster than the size of the blocks in U_{k+1} . \square

6.5. Do Suprema Exist?

In analogy with the situation of the famous degrees of unsolvability [38, 3], one would expect that the interleaving (zip) of two streams yields their supremum. More precisely, one would expect that $\text{zip}(\sigma, \tau)^\diamond$ is the supremum of the degrees σ^\diamond and τ^\diamond . However, in the FST-hierarchy this question is much more complicated.

By Proposition 26 every pair of degrees has an upper bound. For Turing degrees it is known that every pair of degrees has a *least* upper bound, a *supremum*. This raises the question whether the same holds for the FST hierarchy.

We conjecture that the answer to this question is negative:

Conjecture 43. *There exist $\sigma, \tau \in \mathcal{E}^\diamond$ without supremum.*

Let us briefly substantiate this conjecture. We define operations on streams:

$$\begin{aligned} \text{zip}_{1,1}(x : \sigma, y : \tau) &= x : y : \text{zip}_{1,1}(\sigma, \tau) \\ \text{zip}_{1,2}(x : \sigma, y : z : \tau) &= x : y : z : \text{zip}_{1,2}(\sigma, \tau) \end{aligned}$$

Note that $\text{zip}_{1,1}$ is equivalent with the earlier defined zip .

Let σ, τ be streams, and define $\gamma_1 = \text{zip}_{1,1}(\sigma, \tau)$, and $\gamma_2 = \text{zip}_{1,2}(\sigma, \tau)$. Then obviously γ_1 and γ_2 are upper bounds for $\{\sigma, \tau\}$. If $\{\sigma, \tau\}$ has a supremum v , then v must be a common reduct of γ_1 and γ_2 and an upper bound for $\{\sigma, \tau\}$, that is, $\gamma_1 \triangleright v \triangleleft \gamma_2$ and $\sigma \triangleleft v \triangleright \tau$. However, γ_1 and γ_2 contain the information of the streams σ and τ at different speeds. That is, neighbouring digits of γ_1 have unbounded distance in γ_2 and vice versa. Hence, information from a bounded size area of the common reduct v has unbounded distance in either γ_1 or γ_2 . It appears implausible if not impossible that a transducer with finite memory can perform such a transformation. However, for particular streams σ and τ the common reduct v may very well exist. Therefore, the streams σ and τ have to be chosen carefully. For example, σ and τ should be incomparable, that is, $\sigma \not\preceq \tau$ and $\tau \not\preceq \sigma$. Even for streams with this property, it is not excluded that $\text{zip}_{1,1}(\sigma, \tau) \triangleright \text{zip}_{1,2}(\sigma, \tau)$ or $\text{zip}_{1,2}(\sigma, \tau) \triangleright \text{zip}_{1,1}(\sigma, \tau)$, which should be excluded as well.

The operations $\text{zip}_{1,1}$ and $\text{zip}_{1,2}$ are only examples for constructing upper bounds. There are various other constructions for obtaining upper bounds. A possible line of attack for proving Conjecture 43 is as follows:

- (i) Choose streams σ and τ .
- (ii) Choose upper bounds γ_1, γ_2 of $\{\sigma, \tau\}$ such that the set of common reducts $\mathcal{C} = \{v \mid \gamma_1 \triangleright v\} \cap \{v \mid \gamma_2 \triangleright v\}$ of γ_1 and γ_2 has a manageable structure.

(iii) Show that for no stream $v \in \mathcal{C}$ we have $v \triangleright \sigma$ and $v \triangleright \tau$.

The crucial point is (ii), that is, finding upper bounds with a manageable set of common reducts.

6.6. Recurrence

As a final observation in our initial investigation of the FST-hierarchy we show that the property of ultimate recurrence, defined below, is invariant under reducibility \triangleright . Such invariances are useful for discriminating degrees σ^\diamond and τ^\diamond , that is, proving that $\sigma \not\triangleright \tau$ or $\tau \not\triangleright \sigma$.

Definition 44. A stream σ is called *recurrent* if every factor (finite subword) w of σ occurs infinitely often in σ . A stream σ is called *ultimately recurrent* if for some $n \in \mathbb{N}$ the suffix $\sigma|_{\geq n}$ is recurrent.

A simple example shows that the property of being ‘recurrent’ is not invariant under stream transduction. Let $\sigma = 00000\dots$ and $\tau = 100000\dots$. Then obviously $\sigma \triangleright \tau$, but σ is recurrent while τ is not.

However, being ‘ultimately recurrent’ turns out to be an invariance. We give a slightly different characterisation of recurrent streams:

Lemma 45. *A stream σ is recurrent if and only if for all prefixes w of σ we have that w occurs infinitely often in σ . \square*

We now show that ultimate recurrence is preserved by FSTs. Bult [6] derived a similar result for Mealy machines. Other related results have been obtained in [32, 35]: preservation of almost periodicity and of strong almost periodicity (= uniform recurrence) by FSTs, respectively.

Theorem 46. *If $\sigma \triangleright \sigma'$ and σ is ultimately recurrent, then σ' is ultimately recurrent.*

Proof. Let $\sigma, \tau \in \mathbf{2}^\omega$, and $A = \langle Q, q_0, \delta, \lambda \rangle$ an FST such that $\sigma \triangleright_A \tau$ and σ is an ultimately recurrent stream. Let $n_0 \in \mathbb{N}$ such that $\sigma|_{\geq n_0}$ is recurrent.

For $w \in \mathbf{2}^*$, let $\text{Pos}(w)$ denote the set of positions of w in σ , that is, $\text{Pos}(w) = \{n \mid w \sqsubset \sigma|_{\geq n}\}$. Moreover, let $Q_\infty(w) \subseteq Q$ be the set of states $q \in Q$ such that for infinitely many positions $n \in \text{Pos}(w)$ we have $\delta(\sigma|_{< n}) = q$.

Note that:

- (i) $Q_\infty(w) \subseteq Q_\infty(w')$ whenever $w' \sqsubseteq w$, and
- (ii) $Q_\infty(w) \neq \emptyset$ for every factor w of $\sigma|_{\geq n}$ by the Pigeonhole principle.
- (iii) As a consequence of (i) and (ii), for every $n' \geq n$, there exists a state $q \in Q$ such that $q \in Q_\infty(w)$ for every $w \sqsubset \sigma|_{\geq n'}$.

We determine an index $m \in \mathbb{N}$ such that for all prefixes $w \sqsubset \sigma|_{\geq m}$ it holds that $\delta(\sigma|_{< m}) \in Q_\infty(w)$. Then, it follows that $\lambda(\delta(\sigma|_{< n}), \sigma|_{\geq n})$ is recurrent by Lemma 45, and hence $\lambda(\sigma) = \lambda(\sigma|_{< n}) \cdot \lambda(\delta(\sigma|_{< n}), \sigma|_{\geq n})$ is ultimately recurrent.

To determine m , we employ the following iterative algorithm with the invariant $Q_i = Q_\infty(p_i)$ for all $i \in \mathbb{N}$. We start with $m_1 = n_0$, $p_1 = \epsilon$, and $Q_1 = Q_\infty(p_1)$. For $k = 1, 2, \dots$, we define:

- Let $Q_{k+1} = \bigcap \{Q_\infty(p) \mid p \sqsubset \sigma|_{\geq m_k}\}$.
 Note that $\emptyset \neq Q_{k+1}$ by (iii), and $Q_{k+1} \subseteq Q_k$ by (i) since $p_k \sqsubset \sigma|_{\geq m_k}$ with $Q_\infty(p_k) = Q_k$.
- Let p_{k+1} be a prefix of $\sigma|_{\geq m_k}$ such that $Q_\infty(p) = Q_{k+1}$.
 Such a prefix exists as a consequence of (i) and $\emptyset \neq Q_{k+1}$.
- Let $m_{k+1} \in \text{Pos}(p_{k+1})\sigma$ such that $m_{k+1} \geq m_k$ and for every occurrence of p_{k+1} at positions $n \geq m_{k+1}$ in σ it holds $\delta(\sigma|_{\geq n}) \in Q_{k+1}$.

We stop the iterative process as soon as we reach $k \in \mathbb{N}$ such that $Q_{k+1} = Q_k$. Then we define $m = m_k$. The construction is guaranteed to terminate since for all $k' \geq 1$ we have $Q_{k'+1} \subseteq Q_{k'}$.

We check that m has the desired properties. Let $p \sqsubset \sigma|_{\geq m}$. If $p \sqsubset p_k$ then $\emptyset \subseteq Q_\infty(p_k) \subseteq Q_\infty(p)$ and hence $\delta(\sigma|_{< m}) \in Q_\infty(p)$ since $\delta(\sigma|_{< m}) \in Q_\infty(p_k)$. Otherwise if $p_0 \sqsubseteq p$ then $\delta(\sigma|_{< m}) \in Q_\infty(p_k) = Q_\infty(p)$. □

7. Primality of Morse

Of special interest in the FST-hierarchy is the location of well-known streams. In particular, we are intrigued by the wonderful Thue–Morse sequence $M = 01101001 \dots$ and we wonder whether the degree of M is prime.

Question 47. Is the degree M^\diamond prime?

In order to gain insight in this problem, we may consider some non-trivial (i.e., not ultimately periodic) FST-reducts of M and then try to find the way back to M .

As evidence in favor of an affirmative answer to Question 47 one can for instance establish that all arithmetical subsequences of M can be transformed back by some FST to M again: Let $\sigma_{an+b} = (\sigma(an+b))_{n \geq 0}$ denote the subsequence of $\sigma \in A^\omega$ indexed by the arithmetic progression $b, a+b, 2a+b, \dots$:

Proposition 48. *Every arithmetic subsequence of M is equivalent to M :*

$$M \diamond M_{an+b}, \quad \text{for all } a \geq 0, b \geq 1.$$

Proof. (A more detailed proof of Proposition 48 is given in [41].) Let $a \geq 0$ and $b \geq 1$. We define a morphism by $h(0) = 01$ and $h(1) = 10$. Then we have $M = h(M)$. Thus, for every $k \in \mathbb{N}$, M can be viewed as a sequence $M = h^k(0)h^k(1)h^k(1)h^k(0)\dots$ of blocks $h^k(0)$ and $h^k(1)$ of length 2^k . We choose k such that $a \leq 2^k$. If we number the blocks d_0, d_1, \dots , then beginning from the block $d_{\lfloor b/2^k \rfloor}$, the sequence M_{an+b} contains at least one bit from every block. Note that every block d_j is completely determined by an index $0 \leq i < 2^k$ in the block and the bit $d_j(i)$ at position i . A finite stream transducer can reconstruct M from M_{an+b} employing that the sequence $(an+b)_{n \in \mathbb{N}}$ is periodic modulo 2^k , that is, the indexes of the picked digits within each block are periodic. \square

We generalize and extend the question of primality of M to sequences formed by sequences $(v_p(n) \bmod 2)_{n \geq 1}$ where $v_p(n)$ is the p -adic valuation of n (the multiplicity of p in the prime factorisation of n). The Morse sequence M is FST-equivalent to such a sequence, namely the period doubling sequence PD, which is defined as 2-adic valuation modulo 2. Thus M is prime if and only if PD is.

Definition 49. We write $\mathbf{P} = \{2, 3, \dots\}$ for the set of prime numbers. Let $p \in \mathbf{P}$. The p -adic valuation $v_p(n)$ of a number $n \geq 1$ is defined by:

$$v_p(n) = \max \{a \mid p^a \text{ divides } n\}$$

The sequence generated by $A \subseteq \mathbf{P}$, which we denote by \mathbf{v}_A , is defined as follows:

$$v_A(n) = \left(\sum_{p \in A} v_p(n) \right) \bmod 2, \quad \text{for all } n \geq 1.$$

A sequence $\sigma \in \mathbf{2}^\omega$ such that $\sigma = \mathbf{v}_A$ for some $A \subseteq \mathbf{P}$ is called *prime generated*.

For instance the period doubling sequence is defined as $\text{PD} = \mathbf{v}_{\{2\}}$. We mention without proof that sequences generated by a singleton prime are morphic and Toeplitz words.

Question 50. Do there exist sets $A \subseteq \mathbf{P}$ such that the degree of \mathbf{v}_A is prime?

One may also wonder how the sequences relate to each other. We have seen that the Mephisto Waltz $W = 001001110\dots$ and the Sierpiński stream $S = 001111000110000110\dots$ belong to the same degree in the FST-hierarchy, $W \diamond S$ just like $M \diamond \text{PD}$. What about M and S , can one be reduced to the other? It is not hard to show that $\delta(W)$ is prime generated; more precisely:

$$\delta(W) = \mathbf{v}_A \quad \text{where} \quad A = \{p \in \mathbf{P} \mid p \equiv 2 \pmod{3}\} \cup \{3\}$$

Note that A is an infinite set of primes by Dirichlet's Theorem [13].

We conjecture that for every $A \subseteq \mathbf{P}$ the sequence \mathbf{v}_A belongs to a unique degree in the FST-hierarchy:

Conjecture 51. *For all sets $A, B \subseteq \mathbf{P}$ the following implication holds:*

$$\mathbf{v}_A \triangleright \mathbf{v}_B \implies A = B$$

If this is true, M and S are incomparable.

8. Conclusions and Questions

Finite state transducers capture the intuition of infinite pattern. Summarising, the hierarchy of degrees of streams, generated by FSTs, is *fine grained* as FSTs can do only simple transformations, *robust* under the exchange of an arbitrary finite amount of elements in the streams, prepending or cutting-off of prefixes, change of encoding, etc., and captures the *infinitary information content* in the sense of invariant infinite patterns. The main idea leading to this hierarchy of degrees seems to us both simple and necessary to investigate. Remarkably, it seems to have remained unexplored.

We have made a first step in the study on the FST-hierarchy, considering a few elementary properties like well-foundedness, the existence of maximal degrees, prime degrees, Most of all, we have encountered questions, the most interesting of which we want to mention here:

- (i) Is the degree of M (Thue–Morse sequence) prime?
- (ii) How many prime streams degrees exist?
- (iii) How do the degrees of some well-known streams compare? For example, are M (Morse) and S (Sierpiński) of the same degree?
- (iv) How to prove non-reducibility $\sigma \not\preceq \tau$? An especially interesting instance: how to prove non-reducibility between morphic streams?
- (v) Is every prime generated sequence \mathbf{v}_p with $p \in \mathbf{P}$ prime? Note that this would immediately imply (i).
- (vi) Do the structures displayed in Figure 14 exist?

In this paper, we have employed *pumping lemma*-like arguments to prove primality and non-reducibility of certain streams. However, these arguments apply only for streams with a very regular structure, for example for streams built from blocks of the form 10^n of growing size n . It seems challenging to prove primality and non-reducibility of streams that avoid repeating patterns, for example morphic streams like the Morse sequence M.

It is inspiring to compare streams and fractal curves; from fractal curves we obtain interesting new streams such as the stream S, generating the Sierpiński triangle. Can we find criteria for streams which guarantee that the turtle drawings

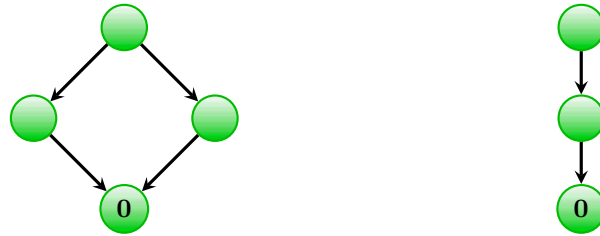


Figure 14: Possible structures in the FST-hierarchy: a diamond, and a line. The arrows $S \rightarrow T$ mean $S \triangleright T$. Using transitivity of \triangleright we leave some arrows implicit. Moreover, we assume that if S is a degree and $S \triangleright T$, then T is depicted as well. In particular there are no intermediate degrees between two displayed nodes connected by an arrow.

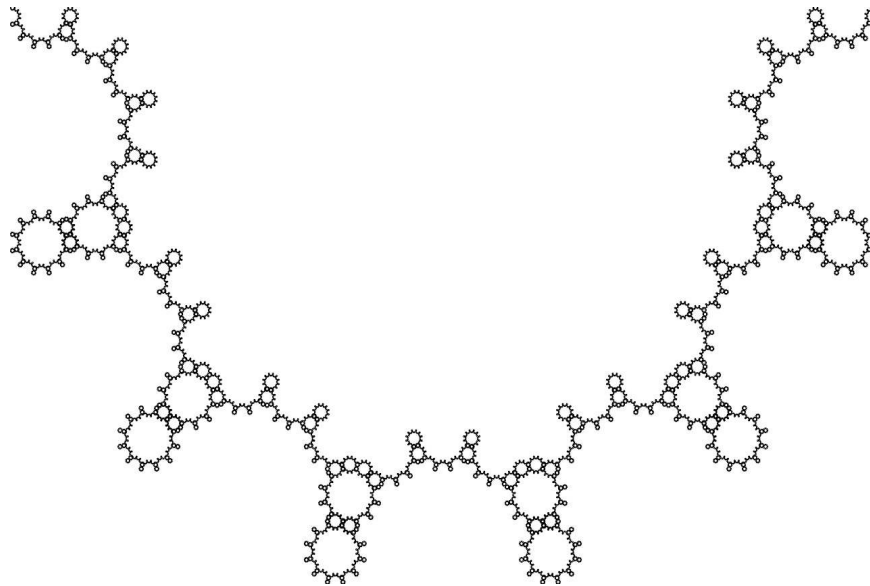


Figure 15: The necklace curve generated by the sequence $v_{\{7\}}$ using the instructions: turn $\pi/6$ radians left on reading 0 and turn $7\pi/6$ right on reading 1. The figure displays the first 7^7 steps.

converge towards a fractal curve? Can fractal curves be used as a tool to distinguish degrees? As an illustration we mention the prime generated sequence $v_{\{7\}}$ (see Section 7) which generates the ‘necklace’ fractal displayed in Figure 15.

Acknowledgements. We thank Frits Dannenberg for his interest in this subject, and refer to his master’s thesis [10] which in particular contains some more general

expressions for characterizing FST-reducts of sequences like II. We thank Alexander Belov for his encouraging comments on our work and making us aware of his own work on the hierarchy that arises from considering Mealy machines (see Related work on page 4). We thank the anonymous referees from prompting us to rewrite and clarify some of the proofs in the appendix.

References

- [1] J.-P. Allouche and R. Bacher. Toeplitz Sequences, Paperfolding, Towers of Hanoi, and Progression-Free Sequences of Integers. *L'Enseignement Mathématique*, 38:315–327, 1992.
- [2] J.-P. Allouche and J. Shallit. *Automatic Sequences: Theory, Applications, Generalizations*. Cambridge University Press, New York, 2003.
- [3] K. Ambos-Spies and P.A. Fejer. Degrees of Unsolvability, 2006.
- [4] Alexander Belov. Some Algebraic Properties of Machine Poset of Infinite Words. *ITA*, 42(3):451–466, 2008.
- [5] V. Berthé. *Sequences Of Low Complexity: Automatic And Sturmian Sequences*. Course notes.
- [6] J. Bult. The Lattice of Machine Invariant Sets and Subword Complexity. *CoRR*, abs/cs/0502064, 2005.
- [7] A. Carpi. Repetitions in the Kolakovski Sequence. *Bull. of the EATCS*, 50:194–196, 1993.
- [8] V. Chvátal. Notes on the Kolakoski Sequence. Technical Report 93-84, DIMACS, 1994.
- [9] K. Culik II, J. Karhumäki, and A. Lepistö. Alternating Iteration of Morphisms and the Kolakovski Sequence. In G. Rozenberg and A. Salomaa, editors, *Lindenmayer Systems*, pages 93–103. Springer, 1992.
- [10] F.G.W. Dannenberg. Symbolic Dynamics and Automata Theory. Master's thesis, Delft University of Technology, 2011.
- [11] F.M. Dekking. Constructies voor 0-1-Rijen met Strikt Ergodische Afgesloten Baan. Master's thesis, Universiteit van Amsterdam, 1974.
- [12] F.M. Dekking. What Is the Long Range Order in the Kolakoski Sequence? In R.V. Moody and J. Patera, editors, *The Mathematics of Long-Range Aperiodic Order (Waterloo, ON, 1995)*, volume 489 of *NATO ASI Series, Series C: Mathematical and Physical Sciences*, pages 115–125. Dordrecht, The Netherlands, 1997. Kluwer Academic Publishers.
- [13] P. G. L. Dirichlet. Beweis des Satzes, daß jede unbegrenzte arithmetische Progression, deren erstes Glied und Differenz ganze Zahlen ohne gemeinschaftlichen Factor sind, unendlich viele Primzahlen enthält. *Abhand. Ak. Wiss. Berlin*, pages 45–81, 1837.
- [14] G. Edgar. *Measure, Topology, and Fractal Geometry*. Springer, 2nd edition, 2008.
- [15] A. Ehrenfeucht, K.P. Lee, and G. Rozenberg. Subword Complexities of Various Classes of Deterministic Developmental Languages Without Interaction. *Theoretical Computer Science*, 1:59–75, 1975.
- [16] A. Ehrenfeucht and G. Rozenberg. Subword Complexity of Square-Free D0L Languages. *Theoretical Computer Science*, 16(1):25–32, 1981.
- [17] J. Endrullis, C. Grabmayer, and D. Hendriks. Data-Oblivious Stream Productivity. In *Proc. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2008)*, pages 79–96, 2008.
- [18] J. Endrullis, C. Grabmayer, D. Hendriks, A. Isihara, and J.W. Klop. Productivity of Stream Definitions. *Theoretical Computer Science*, 411(4-5):765–782, 2010.

- [19] J. Endrullis, C. Grabmayer, D. Hendriks, and J.W. Klop. Infinite Streams. In *Nieuwsbrief van de NVTI (Newsletter of the Dutch Association for Theoretical Computer Science)*, pages 39–48. 2009. <http://www.nvti.nl/Newsletter/Nieuwsbrief2009.pdf>.
- [20] J. Endrullis, D. Hendriks, and J.W. Klop. Let’s Make a Difference! *CoRR*, abs/0911.1004, 2009.
- [21] R.A. Holmgren. *A First Course in Discrete Dynamical Systems*. Springer Universitext, 2nd edition, 1996.
- [22] J.E. Hutchinson. Fractals and Self-Similarity. *Indiana Univ. Math. J.*, 30:713–747, 1981.
- [23] K. Jacobs. *Invitation to Mathematics*. Princeton University Press, 1992.
- [24] K. Jacobs and M. Keane. 0-1-Sequences of Toeplitz Type. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 13(2):123–131, 1969.
- [25] M. Keane. Generalized Morse Sequences. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 10(4):335–353, 1968.
- [26] J.W. Klop. A Wonderful Stream. In F. de Boer et al., editor, *Liber Amicorum Jaco de Bakker*, pages 125–142. CWI, 2002.
- [27] W. Kolakowski. Self Generating Runs. *Amer. Math. Monthly*, 72, 1965. Problem 5304.
- [28] M. Li and P.M.B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 2nd edition, 2008.
- [29] M. Lothaire. *Algebraic Combinatorics on Words*. Cambridge University Press, 2002.
- [30] B. Löwe. Complexity Hierarchies Derived from Reduction Functions. In *Classical and New Paradigms of Computation and their Complexity Hierarchies*, volume 23 of *Trends in Logic*, pages 1–14. Springer, 2004.
- [31] J. Ma and J.A. Holdener. When Thue–Morse meets Koch. In *Fractals: Complex Geometry, Patterns, and Scaling in Nature and Society*, volume 13, pages 191–206, 2005.
- [32] A.A. Muchnik, A.L. Semenov, and M. Ushakov. Almost periodic sequences. *Theoretical Computer Science*, 1-3(304):1–33, 2003.
- [33] G. Paun. How much Thue is Kolakowski? *Bull. of the EATCS*, 49:183–185, 1993.
- [34] H. Peitgen, H. Jürgens, and D. Saupe. *Chaos and Fractals*. New Frontiers of Science. Springer, 2nd edition, 2004.
- [35] Y. Pritykin. Strongly Almost Periodic Sequences under Finite Automata Mappings. *CoRR*, abs/cs/0605026, 2006.
- [36] N. Pytheas Fogg. *Substitutions in Dynamics, Arithmetics and Combinatorics*, volume 1794 of *Lecture Notes in Mathematics*. Springer, 2002.
- [37] J. Sakarovitch. *Elements Of Automata Theory*. Cambridge, 2003.
- [38] J.R. Shoenfield. *Degrees of Unsolvability*. North-Holland, Elsevier, 1971.
- [39] N.J.A. Sloane. Online Encyclopedia of Integer Sequences, 2010. <http://oeis.org/classic/>.
- [40] R. Steacy. Structure in the Kolakoski Sequence. *Bull. of the EATCS*, 59:173–181, 1996.
- [41] S. Stern. The Thue–Morse Sequence. Master’s thesis, Vrije Universiteit Amsterdam, 2008.
- [42] Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

Appendix A. Proof of Theorem 36: Π is Prime

Consider words $w = 0^n$, $w' = 0^m$ with $|Q| < n < m$ such that the difference of the lengths $(m - n)$ is a multiple of Z_A . Let L be the final zero-loop in the run of A on w . Then the run of A on w' is an extension of the run on w with (whole) multiples of L . Lemma 29 lemma employs this finding in the consideration of words of the form $v 0^*$.

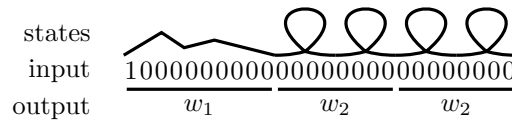


Figure 16: FST reading a word of the form 10^* .

Proof of Lemma 29. Since $n > |Q|$ there exist $k, p \in \mathbb{N}$ with $k + p \leq n$ such that $\delta(q, v0^k) = \delta(q, v0^{k+p})$, that is, we have a repetition of states. Let $k, p \in \mathbb{N}$ be minimal with these properties. Then p is the length of a zero-loop in A , and hence $Z_A = m \cdot p$ for some $m \in \mathbb{N}$. It follows that $\delta(q, v0^{k'}) = \delta(q, v0^{k'+p})$ for every $k' \geq k$, and as a consequence

$$\delta(q, v 0^{n+i \cdot Z_A}) = \delta(q, v 0^{n+i \cdot m \cdot p}) = \delta(q, v 0^n)$$

Define $w_1 = \lambda(q, v 0^n)$, $q' = \delta(q, v 0^n)$ and $w_2 = \lambda(q', 0^{Z_A})$; see Figure 16. Then

$$\lambda(q, v 0^{n+i \cdot Z_A}) = w_1 w_2^i$$

since $\delta(q', 0^{Z_A}) = q'$ as shown above. □

In the sequel we will employ Lemma 29 for the special case of $v = 1$.

The Structure of Reducts of Π

We give a few examples, to illustrate the Proposition 30 and Theorem 36.

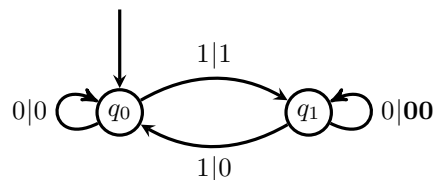


Figure 17: FST deleting every second 1, doubling the zeros in even blocks.

Example 52. The FST from Figure 17 removes every second 1, and doubles the zeros in even blocks 10^* . That is, it reduces Π to:

$$\begin{aligned} \Pi_2 &= 1001000000001000000000000001\dots \\ &= 1\ 0^2\ 10^8\ 10^{14}\ 1\dots \\ &= \prod_{i=0}^{\infty} (1\ (0000)^i \cdot 0\ 0\ (00)^i) = \prod_{i=0}^{\infty} \prod_{j=0}^1 p_j \cdot c_j^i \end{aligned}$$

where $p_0 = 1$, $c_0 = 0000$, $p_1 = 00$ and $c_1 = 00$.

We can transform Π_2 back to Π by compressing blocks of zeros $0^n \mapsto 0^{\frac{n-2}{6}}$. Again, we can construct an FST performing this compression by Lemma 35.

Let us consider a slightly slightly more involved example:

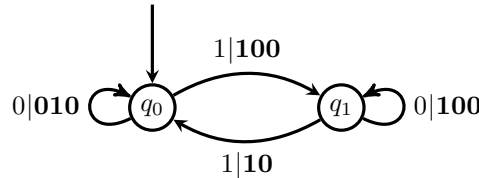


Figure 18: FST transforming Π .

Example 53. The FST from Figure 18 transforms Π into:

$$\begin{aligned} \Pi_3 &= 1001001010010010010010010010010010010010010010010010010010010010\dots \\ &= \prod_{i=0}^{\infty} (100\ (100100)^i \cdot 10\ 010\ (010010)^i) = \prod_{i=0}^{\infty} \prod_{j=0}^1 p_j \cdot c_j^i \end{aligned}$$

where $p_0 = 100$, $c_0 = 100100$, $p_1 = 10010$ and $c_1 = 010010$.

The stream can be transformed back to Π . The stream Π_3 is not ultimately periodic. In particular, the periodicity is broken by the occurrences of 101, and we can construct an FST that recognizes these occurrences. A simple calculation shows that the distance between 101 grows linearly, and hence we can reconstruct Π by Lemma 35.

We remark that a slight modification of the FST from Figure 18, as shown in Figure 19, would result in the (ultimately) periodic stream $(010)^\omega$ which cannot be transformed back to Π .

Proof of Proposition 30. Let $A = \langle Q, q_0, \delta, \lambda \rangle$ be an FST such that $\Pi \triangleright_A \sigma$. Define $f_\Pi(n)$ as the position of the $(n + 1)$ -th 1 in Π , that is:

$$f_\Pi(n) = \frac{(n + 1) \cdot n}{2}$$

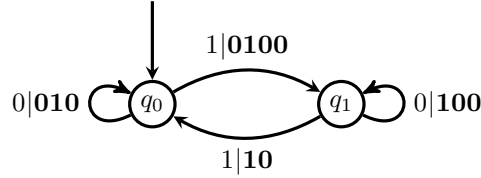


Figure 19: FST transforming Π to a periodic stream.

By the Pigeonhole Principle there exist $n_1, n_2 \in \mathbb{N}$ such that $|Q| < n_1 < n_2$, and $\delta(\Pi|_{<f_{\Pi}(n_1)}) = \delta(\Pi|_{<f_{\Pi}(n_2)})$, and $n_1 \equiv n_2 \pmod{Z_A}$. In other words, we consider two blocks 10^{n_1} and 10^{n_2} in Π that the automaton A enters in the same state and whose length difference is a multiple of Z_A .

Let $m \in \mathbb{N}$ such that $n_2 - n_1 = m \cdot Z_A$. Note that $\Pi|_{\geq f_{\Pi}(n)}$ begins with the prefix 10^n (followed by a 1) for every $n \in \mathbb{N}$. For every $i \in \mathbb{N}$, we have $n_2 + i = n_1 + i + m \cdot Z_A$. As a consequence, by Lemma 29 together with induction on $i \in \mathbb{N}$ we obtain: $\delta(\Pi|_{<f(n_1+i+m \cdot Z_A)}) = \delta(\Pi|_{<f(n_1+i)})$ for all $i \in \mathbb{N}$. Define $q_j = \delta(\Pi|_{<f(n_1+j)})$ for $j = 0, \dots, m \cdot Z_A$. Then $\delta(\Pi|_{<f(n_1+j+k \cdot m \cdot Z_A)}) = q_j$ for all $j \in \{0, \dots, m \cdot Z_A\}$ and $k \in \mathbb{N}$, and hence $\lambda(q_j, 10^{n_1+j+k \cdot m \cdot Z_A}) = p_j(v_{j,2}^m)^k$ by Lemma 29, for suitable words $p_j, v_{j,2} \in \mathbf{2}^*$.

Define $w = \lambda(\Pi|_{<f(n_1)})$ and $c_j = v_{j,2}^m$. Then we obtain:

$$\sigma = w \cdot \prod_{k=0}^{\infty} \prod_{j=0}^{m \cdot Z_A} \lambda(q_j, 10^{n_1+j+k \cdot m \cdot Z_A}) = w \cdot \prod_{k=0}^{\infty} \prod_{j=0}^{m \cdot Z_A} p_j \cdot c_j^k$$

which proves the claim. □

Breaking Points of Periodicity

In this section we consider *non-ultimately periodic* streams σ defined by:

$$\sigma = w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^n p_j \cdot c_j^i \tag{1}$$

where $n \in \mathbb{N}$ and $w, p_j, c_j \in \mathbf{2}^*$ are finite words for $j = 0, \dots, n$.

We show that the double product can always be rewritten to a form such that a finite state transducer reading σ can recognize the transition from one factor $p_j \cdot c_j^i$ to the subsequent factor $p_{j+1} \cdot c_{j+1}^i$.

The following definition allows us to unroll the first factor $p_0 \cdot c_0^0$ of the inner product. Thereby the factors of the inner product get rotated, and hence we can bring every factor of the inner product to the front.

Definition 54. By *unrolling* we refer to the transformation from (1) to:

$$\begin{aligned} \sigma &= \underbrace{w \cdot p_0}_{w'} \cdot \prod_{i=0}^{\infty} \left(\underbrace{\left(\prod_{j=0}^{n-1} p_{j+1} \cdot c_{j+1}^i \right)}_{p'_j \cdot c_j^i} \cdot \underbrace{p_0 c_0}_{p'_n} \cdot \underbrace{c_0^i}_{c_n^i} \right) \\ &= w' \cdot \prod_{i=0}^{\infty} \prod_{j=0}^n p'_j \cdot c_j^i \end{aligned}$$

where $w' = w \cdot p_0$, $p_j = p_{j+1}$ and $c_j = c_{j+1}$ for $0 \leq j < n$, and $p'_n = p_0 c_0$ and $c'_n = c_0$.

We present different transformations that allow us to stepwise simplify the product until it has the desired form. The next lemma, for example, allows us to ‘remove’ factors $p_j \cdot c_j^i$ for which $c_j = \varepsilon$:

Lemma 55. *If $n \geq 1$ and $c_0 = \varepsilon$, then we have:*

$$\sigma = w \cdot \prod_{i=0}^{\infty} (p_0 \cdot p_1 \cdot c_1^i \prod_{j=2}^n p_j \cdot c_j^i) = w' \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{n-1} p'_j \cdot c_j^i$$

where $w' = w$, $p'_0 = p_0 \cdot p_1$, $c'_0 = c_1$ and $p'_j = p_{j+1}$, $c'_j = c_{j+1}$ for all $0 \leq j \leq n - 1$. □

We have $c_j \neq \varepsilon$ for some $0 \leq j \leq n$ since otherwise σ would be ultimately periodic. Employing unrolling together with Lemma 55 we can transform (1) to a form where $c_j \neq \varepsilon$ for all $0 \leq j \leq n$. Hence, without loss of generality, we assume $c_j \neq \varepsilon$ for all $0 \leq j \leq n$ in the sequel.

Definition 56. An index $k \in \{0, \dots, n - 1\}$ is said to have *confusion* if we have $c_k^\omega = p_{k+1} \cdot c_{k+1}^\omega$. Likewise, index n has *confusion* if $c_n^\omega = p_0 \cdot c_0^\omega$.

For indexes j with confusion a finite state transducer cannot recognize the transition from the factor $p_j \cdot c_j^i$ to $p_{j+1} \cdot c_{j+1}^i$.

Example 57. Reconsider Example 53 where $p_0 = 100$, $c_0 = 100100$, $p_1 = 10010$ and $c_1 = 010010$. Here the index 0 has confusion since $c_0^\omega = (100)^\omega = p_1 \cdot c_1^\omega$. This is in contrast to the index 1 for which we have $c_1^\omega = (010)^\omega \neq (100)^\omega = p_0 \cdot c_0^\omega$.

We now work towards Lemma 60 stating that there is an index without confusion.

Lemma 58. *Let $x, y \in \{0, 1\}^+$, $v \in \{0, 1\}^*$ such that $x^\omega = vy^\omega$. Then for some $u \in \{0, 1\}^*$ we have $y = uv$ or $v = uy$.*

Proof. From $x^\omega = vy^\omega$ it follows that $x^{|y|}x^\omega = x^{|y|}vy^\omega = vy^{|x|}y^\omega$. Hence we obtain $x^{|y|}v = vy^{|x|}$ from $|x^{|y|}| = |x| \cdot |y| = |y^{|x|}|$. Thus we have a word that has both v and y as a suffix, and so the claim follows. □

Lemma 59. *Let $0 \leq k < n$, $v \in \{0, 1\}^*$ and $m \in \mathbb{N}$ maximal such that $vc_{k+1}^m = p_{k+1}$. Then k has confusion if and only if $c_{k+1} = uv$ for some $u \in \{0, 1\}^*$ such that c_k and vu are powers of the same word.*

Proof. Assume k has confusion, $c_k^\omega = p_{k+1}c_{k+1}^\omega$, and so $c_k^\omega = vc_{k+1}^\omega$. By Lemma 58 we get $c_{k+1} = uv$ for some $u \in \{0, 1\}^*$ (the other case $v = uc_{k+1}$ is excluded by maximality of m). Hence $c_k^\omega = vc_{k+1}^\omega = v(uv)^\omega = (vu)^\omega$ and so $c_k^{|vu|} = (vu)^{|c_k|}$. Then by the second theorem of Lyndon and Schützenberger [2, Theorem 1.5.3], c_k and vu are powers of the same word.

For the other direction, let $c_{k+1} = uv$ for some $u \in \{0, 1\}^*$ such that c_k and vu are powers of the same word. Hence $c_k^\omega = (vu)^\omega = v(uv)^\omega = vc_{k+1}^\omega = p_{k+1}c_{k+1}^\omega$. \square

By the following lemma, there always exists at least one index without confusion since otherwise σ would be ultimately periodic:

Lemma 60. *There exists an index $k \in \{0, \dots, n\}$ without confusion.*

Proof. Assume that all indexes $0 \leq k \leq n$ have confusion. We show that then the stream σ would be ultimately periodic. For this purpose it suffices to show that $v_\ell \sqsubset wp_0c_0^\omega$ for every $\ell \in \mathbb{N}$, where v_ℓ is the following prefix of σ :

$$v_\ell = w p_0c_0^0 \dots p_n c_n^0 p_0c_0^1 \dots p_n c_n^1 \dots p_0c_0^\ell \dots p_n c_n^\ell$$

For $0 \leq i \leq n$ and $0 \leq j \leq \ell$, let $q_{i,j}$ denote the suffix of v starting with $p_i c_i^j$, that is, $q_{i,j} = p_i c_i^j \dots p_n c_n^j \dots p_0 c_0^\ell \dots p_n c_n^\ell$. We show by induction on the length of $q_{i,j}$ that $q_{i,j} \sqsubset p_i c_i^\omega$. For the base case: $q_{n,\ell} = p_n c_n^\ell \sqsubset p_n c_n^\omega$. For the induction step, we distinguish two cases: If $i < n$, then as i has confusion, $c_i^\omega = p_{i+1}c_{i+1}^\omega$ and hence $q_{i+1,j} \sqsubset p_{i+1}c_{i+1}^\omega = c_i^\omega$ by the induction hypothesis, and $q_{i,j} = p_i c_i^j q_{i+1,j} \sqsubset p_i c_i^\omega$. In case $i = n$, we reason analogously. It follows that σ is ultimately periodic with period c_0 . \square

Next we show that all indexes with confusion can be removed:

Lemma 61. *Assume that index 0 has confusion (then $n \geq 1$ by Lemma 60). Then:*

$$\sigma = w \cdot \prod_{i=0}^{\infty} \prod_{j=0}^{n-1} p'_j \cdot c_j^i$$

where

- $p'_0 = p_0 \cdot p_1$,
- $p'_j = p_{j+1}$,
- c'_0 is the suffix of length $|c_0| + |c_1|$ of $c_1^{|c_0|+|c_1|}$, and
- $c'_j = c_{j+1}$ for all $1 \leq j \leq n - 1$.

Proof. Define $v_i = p_0 \cdot c_0^i \cdot p_1 \cdot c_1^i$ for all $i \in \mathbb{N}$. It suffices to show that $p'_0 \cdot c_0^i = v_i$. We proceed by induction on i . The base case $i = 0$ is trivial. For the induction step we show that $v_{i+1} = v_i c'_0$ holds (then by the induction hypothesis $v_{i+1} = v_i c'_0 = p'_0 c_0^i c'_0 = p'_0 c_0^{i+1}$).

By confusion of the index 0 we have $c_0^\omega = p_1 \cdot c_1^\omega$ and hence $c_0^\omega = c_0^j \cdot p_1 \cdot c_1^\omega$ for all $j \in \mathbb{N}$. As a consequence $c_0^j \cdot p_1 \cdot c_1^j \sqsubset c_0^\omega$ and $v_j = p_0 c_0^j \cdot p_1 \cdot c_1^j \sqsubset p_0 c_0^\omega$ for all $j \in \mathbb{N}$. Hence $v_{i+1} = v_i \cdot u_i$ for some $u_i \in \mathbf{2}^*$. We have $|u_i| = |c'_0|$ and so it remains to be shown that u_i is a suffix of $c_1^{|c_0|+|c_1|}$ (just like c'_0).

Clearly we also have that $c_0^j c_0^i p_1 c_1^i \sqsubset p_1 c_1^\omega$ for all $j \in \mathbb{N}$. We take j such that $|c_0^j| > |p_1|$ and $|c_0^{j+i}| \equiv 0 \pmod{|c_1|}$. Hence $c_0^j p_1 c_1^i$ is a suffix c_1^e for some $e \in \mathbb{N}$. Since u_i is a suffix of $c_0^i p_1 c_1^i$ we obtain that u_i is a suffix of $c_1^{|c_0|+|c_1|}$. \square

Again, employing unrolling together with Lemma 61 we can transform (1) to a form where no index $0 \leq k \leq n$ has confusion.

Definition 62. By *unfolding* we refer to the transformation from (1) to:

$$\sigma = \underbrace{\left(w \cdot \prod_{j=0}^n p_j \right)}_{w'} \cdot \prod_{i=0}^{\infty} \underbrace{\prod_{j=0}^n (p_j \cdot c_j)}_{p'_j} \cdot c_j^i$$

where w' and p'_j are as indicated, and $c'_j = c_j$.

The following lemma states that $v^\omega = v_1 \cdot v_2^\omega$ can be decided by inspecting a finite prefix of both streams:

Lemma 63. *Let $v, v_1, v_2 \in \mathbf{2}^*$ such that $v, v_2 \neq \varepsilon$. Then $v^\omega = v_1 \cdot v_2^\omega$ holds if and only if $v^p \sqsubset v_1 \cdot v_2^q$ where $p = |v_1| + |v_2|$ and $q = p \cdot |v|$.*

Proof. Note that $|v^p| \leq |v_1 \cdot v_2^q|$ by the choice of q and $|v_2| \geq 1$. As a consequence, the direction ‘ \Rightarrow ’ follows. For ‘ \Leftarrow ’ let $v^p \sqsubset v_1 \cdot v_2^q$. Then $v^p = v^{|v_1|} \cdot v^{|v_2|}$ and $v_1 \sqsubset v^{|v_1|}$. We have $|v^{|v_2|}| = k \cdot |v_2|$ for some $k \in \mathbb{N}$. Consequently it follows that $v^{|v_1|} \cdot v^{|v_2|} \sqsubset v_1 \cdot v_2^\omega$ implies $v^{|v_1|} \cdot (v^{|v_2|})^\ell \sqsubset v_1 \cdot v_2^\omega$ for all $\ell \in \mathbb{N}$. \square

Using unfolding together with the bounds implied by Lemma 63 we can transform (1) to a form where $p_{j+1} \not\sqsubset c_j^\omega$ for all $0 \leq k \leq n - 1$, and $p_0 \not\sqsubset c_n^\omega$. This immediately implies that there is no confusion.

Proof of Proposition 32. Proposition 32 summarizes the results of this section. \square

Stream Transducers and Linear Compression

Note that the Definition 34 considers only non-empty words w (length $|w| \geq 1$). The reason is that we employ FSTs without ε -transitions, and as a consequence, on the input of an empty word, the output is always empty.

Proof of Lemma 35. Assume $a + b \geq 0$. Let $b = c/d$ with $c, d \in \mathbb{N}$. Define $A = \langle Q, q_0, \delta, \lambda \rangle$ as follows. Let $Q = \{0, 1, \dots, d\}$, $q_0 = 0$, and $\delta(n, x) = n + 1$ for $n \in Q \setminus \{d\}$ and $\delta(d, x) = 1$ for every $x \in \mathbf{2}$. Moreover define:

$$\lambda(0, x) = 0^{f(1)} \qquad \lambda(n, x) = 0^{f(n+1)-f(n)}$$

for every $x \in \mathbf{2}$ and $n \in Q \setminus \{1\}$.

For every non-empty word w we have $1 \leq \delta(w) \leq d$ and $\delta(w) \equiv |w| \pmod d$ (follows immediately by induction on $|w|$). Again by induction on $|w|$ we prove $\lambda(w) = 0^{f(|w|)}$. For $|w| = 1$ the claim is trivial. Let $|w| > 1$. We have $w = w'x$ for some $w' \in \mathbf{2}^*$ and $x \in \mathbf{2}$. Then $\lambda(w) = \lambda(w')\lambda(\delta(w'), x)$ which is by induction hypothesis equivalent to $0^{f(|w|-1)}\lambda(\delta(w'), x)$, and we have

$$\lambda(\delta(w'), x) = 0^{f(\delta(w')+1)-f(\delta(w'))} = 0^{f(|w'|+1)-f(|w'|)}$$

since $f(n + 1) - f(n) = f(m + 1) - f(m)$ whenever $n, m \geq 1$ and $n \equiv m \pmod d$. Consequently $\lambda(w) = 0^{f(|w|-1)}0^{f(|w|)-f(|w|-1)} = 0^{f(|w|)}$.

Let us consider the case $a + b < 0$. The case $a < 0$ and $b = 0$ is covered by the trivial FST consisting of only one state and empty output ε along all edges. Therefore assume $a < 0$ and $b > 0$. Let $n \in \mathbb{N}$ be minimal such that $a + b \cdot n \geq 0$. Let A' be the FST constructed for $a' = a + b \cdot (n - 1)$ and $b' = b$. We define A as extension of A' with states $\{q_1, \dots, q_{n-1}\}$ and $\delta(q_i, x) = q_{i+1}$, $\delta(q_{n-1}, x) = 0$ for every $1 \leq i \leq n$, and $\lambda(q_i, x) = \varepsilon$ for every $1 \leq i < n$ and $x \in \mathbf{2}$. Then after $n - 1$ transitions without output, the automaton A starts behaving as A' : $a + b \cdot (n - 1 + i) = a + b \cdot (n - 1) + b \cdot i = a' + b' \cdot i$. \square