Computational Aspects of Symbolic Dynamics

Part II: The Aubrun-Sablik proof

E. Jeandel

LORIA (Nancy, France)

The theorem

Theorem (Aubrun-Sablik [AS], Durand-Romashchenko-Shen [DRS10])

For every 1D effective subshift S, the 2D subshift :

$$S^{\mathbb{Z}} = \{y | \exists x \in S, \forall i, j, y_{ij} = x_i\}$$

$$S^{\mathbb{Z}} = \{y | \textit{all lines are equal to the same } x \in S\}$$

is sofic.

In this talk, the proof by Aubrun-Sablik, and one application.

The general framework

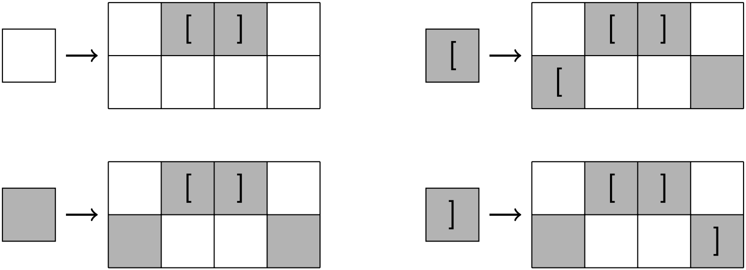
The SFT is divided into two layers

- The first layer contains the word u of $\Sigma^{\mathbb{Z}}$ we want to test. By some simple forbidden patterns, we ensure that u is the same word on each line.
- The second layer contains a construction that tests whether u is valid.

The sofic shift will forget all but the first layer

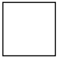
How to simulate a Turing machine

We start from a result of Mozes[Moz89] (every 2D substitution is sofic) and the following substitution :

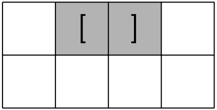


How does this substitution behave ?

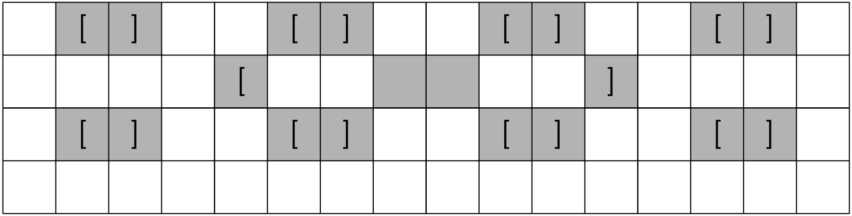
The substitution



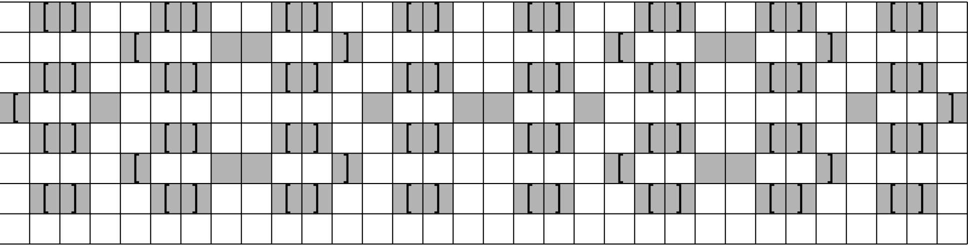
The substitution



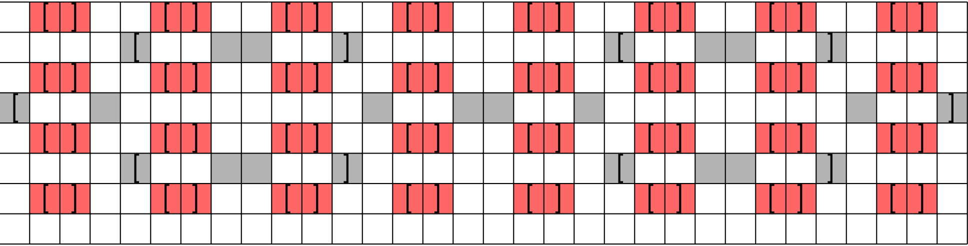
The substitution



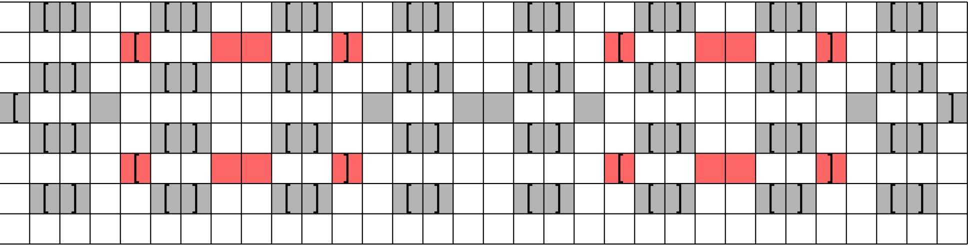
The substitution



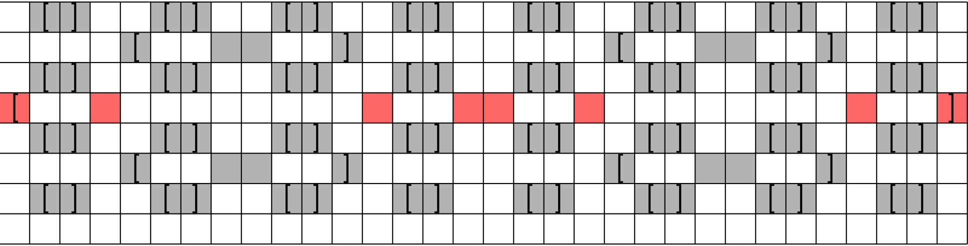
The substitution



The substitution



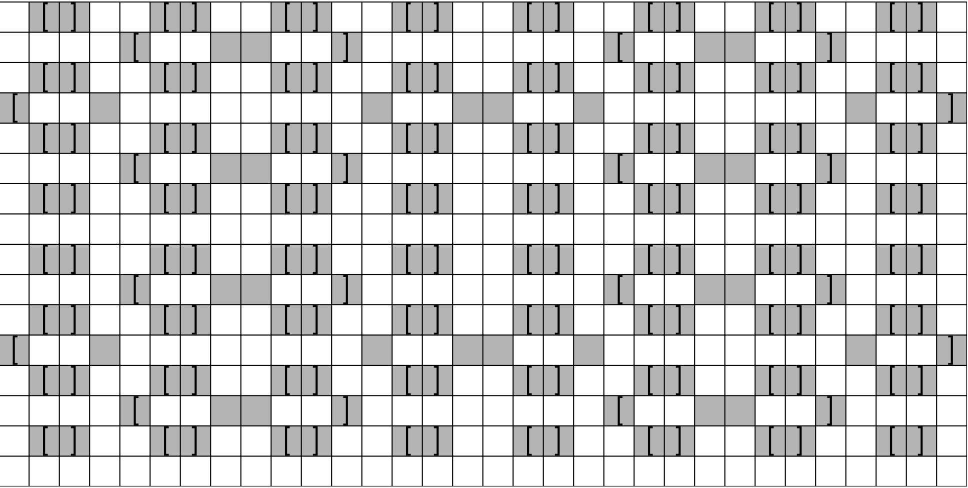
The substitution



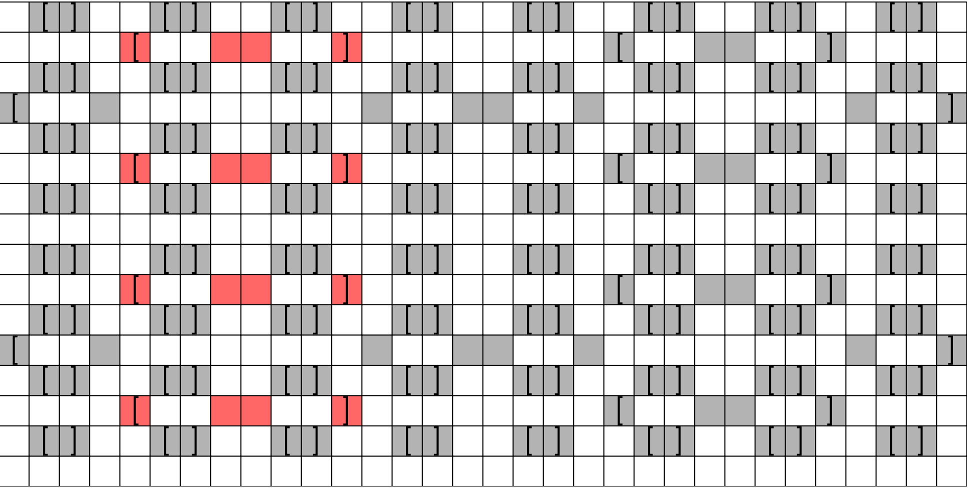
Substitution

- Each* gray square belongs to a vertical strip of gray squares delimited by brackets of a given level n
- Each* white square on the same column/line than a gray square belongs to the same strip, and can be used for communications
- The width of a strip of level n is $O(4^n)$. It contains $O(2^n)$ gray squares.
- The distance between two strips of the same level is $O(2^n)$

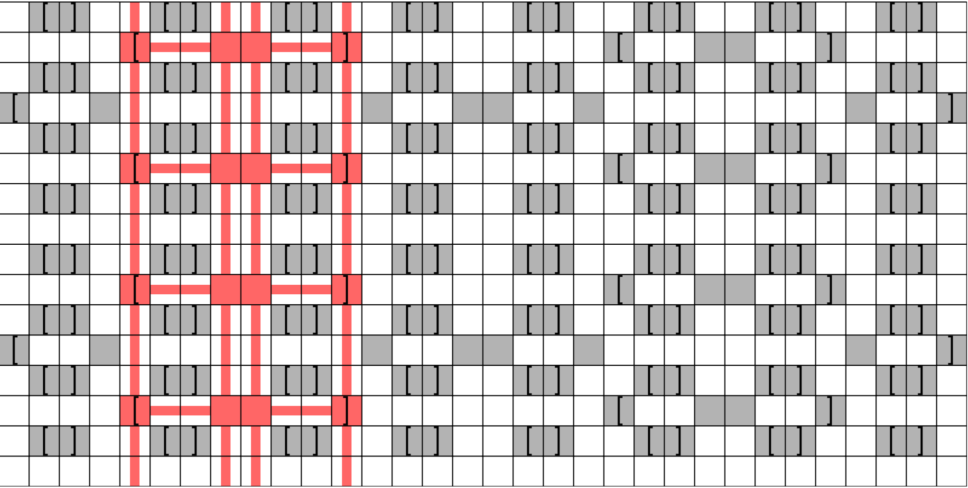
The idea



The idea



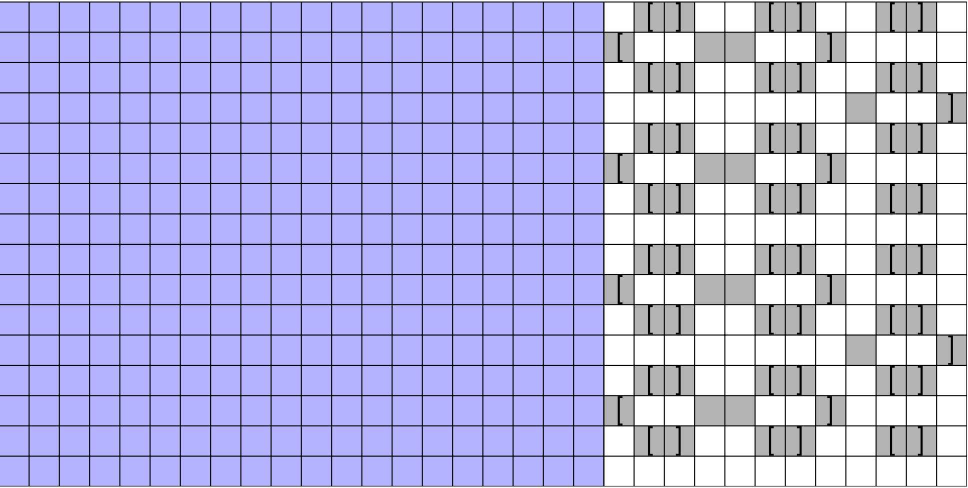
The idea



The idea

- Each strip of size p will be responsible for a zone of size $3p$, and will try to prove that no forbidden word appear in this zone
- (Recall that the first layer contains the same word u on each line)
- The responsibility zone needs to be bigger than the strip so that every finite word appears in some responsibility zone

The responsibility zone



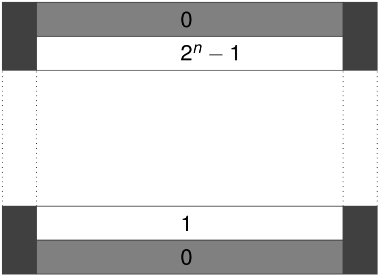
The idea

- Each stripe of size p will have a word u of size $3p$ on input and try to prove that u does not contain any forbidden word.
- We need to encode a Turing machine inside each strip.
- How to compute in a vertical strip ?

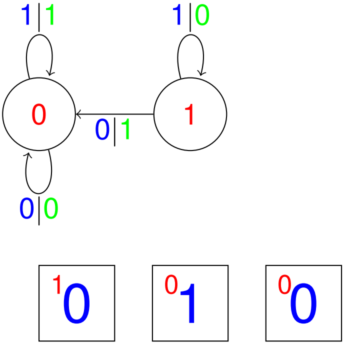
More on how to access the input later

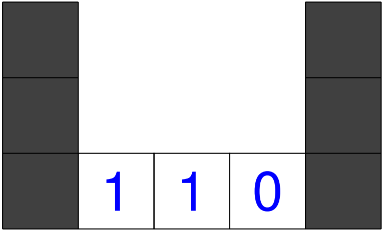
A binary counter

Use a binary counter : Every gray square will contain 0 or 1.

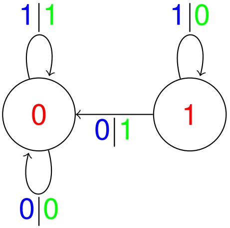


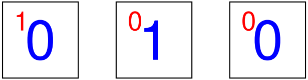
A binary counter

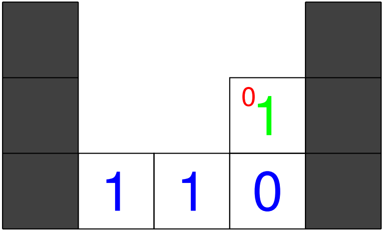




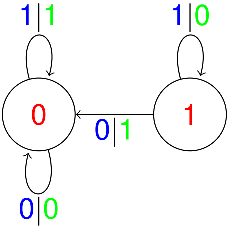
A binary counter

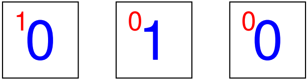


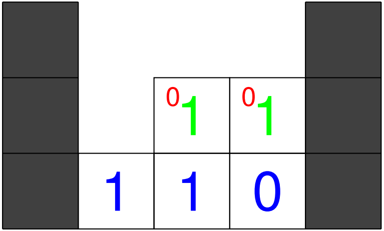




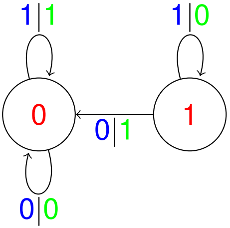
A binary counter

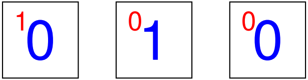


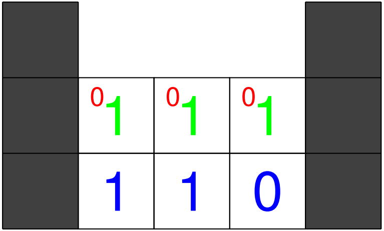




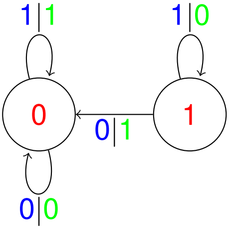
A binary counter

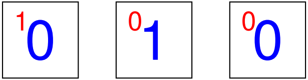


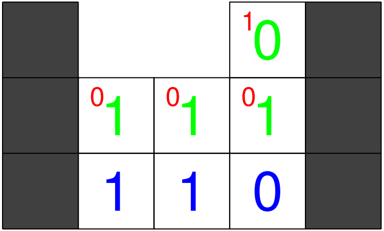




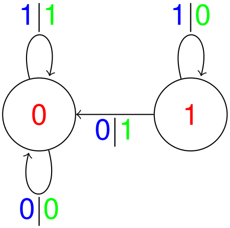
A binary counter

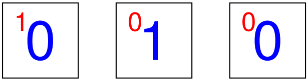


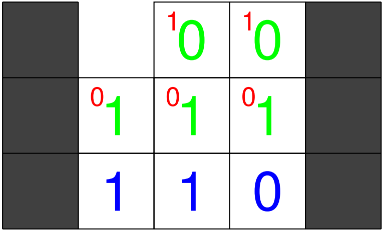




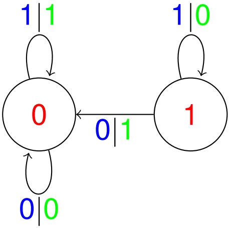
A binary counter

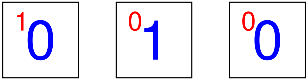


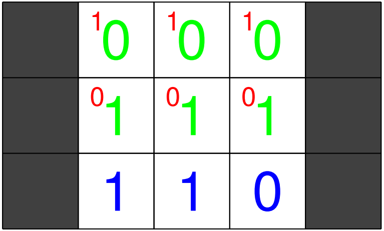




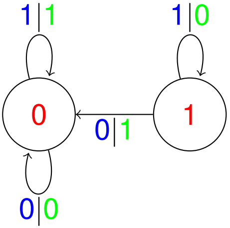
A binary counter

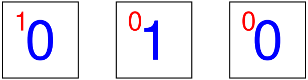


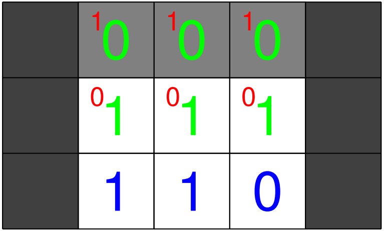




A binary counter







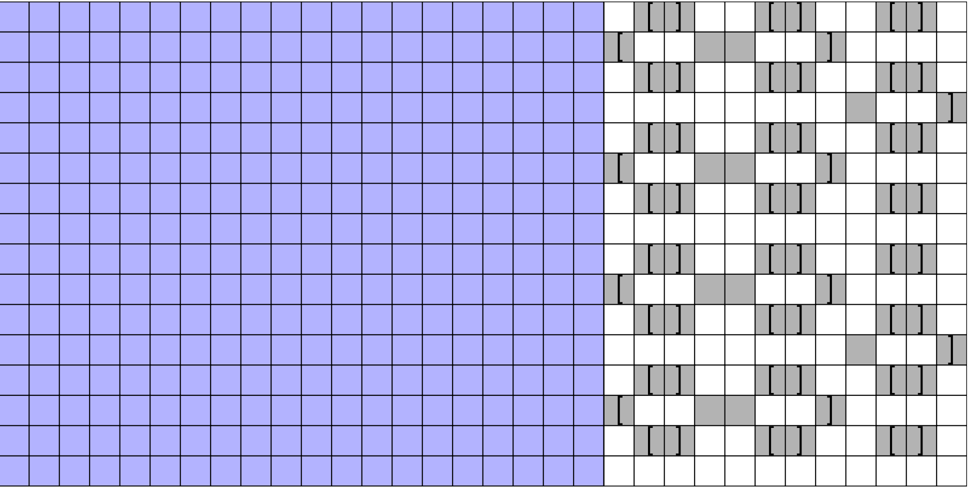
What we obtain

- Each vertical strip of level n is now divided into *rectangles* of size $2^n \times 2^{2^n}$
- We can encode a Turing machine in each rectangle.
- We can use the corner of the rectangle to initialize correctly the Turing machine

Each subword w of u will be tested only for a bounded number of steps by a Turing machine of level n. However w will be tested by machines of arbitrarily large levels, so it's ok.

Last (but big) step

Let's look at the responsibility zone again



How to obtain the input

- When a Turing machine of level n wants to know what symbol is in position i, it might have to *delegate*
  - Either to one of its two neighbours of the same level n
  - Or to one of the four machines of level $n - 1$.

Machines will now have to work for themselves, but also for their neighbour and for their “parent”.

How do machines communicate ?

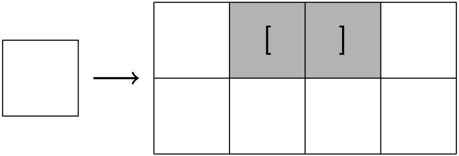
How to communicate

A Turing machine of level n needs to be able to communicate with

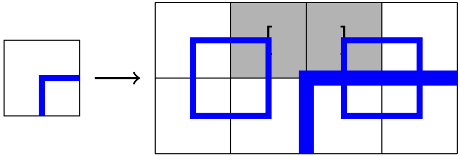
- Its neighbour (easy)
- Machines of level $n - 1$ (harder)

Aubrun and Sablik introduce the brillant idea of a *communication channel*.

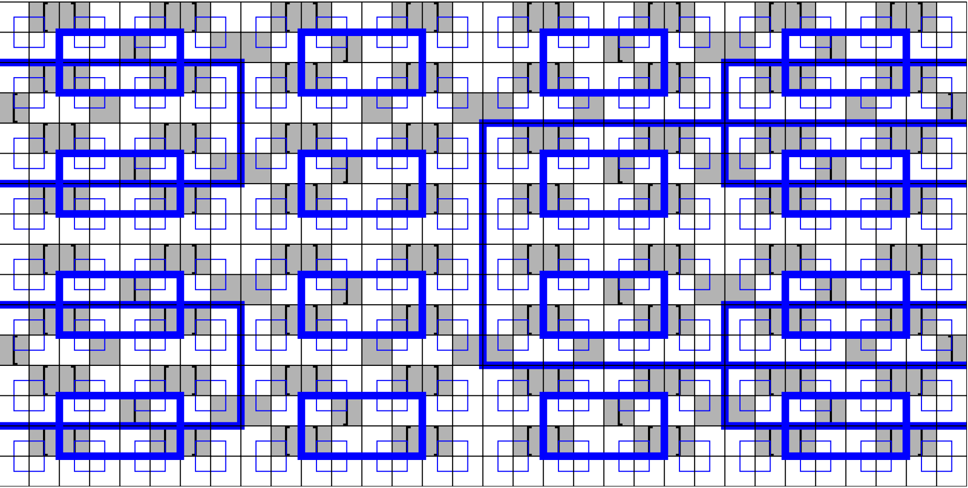
The substitution (again)



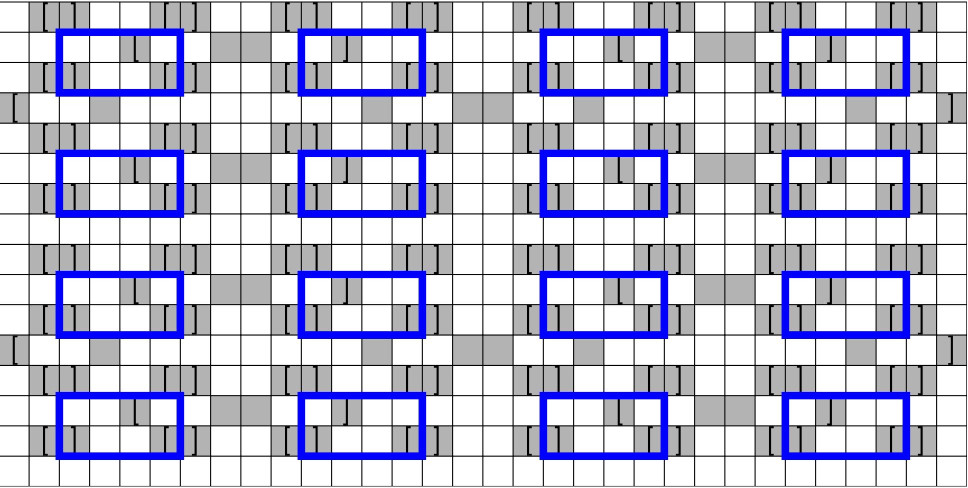
The substitution (again)



Iterating a few times



Iterating a few times



The end

- We can use the rectangles to transmit informations between strips of different levels
- That's basically the end of the proof, up to 25 pages of technical details.

The theorem (again)

Theorem (Aubrun-Sablik, Durand-Romashchenko-Shen)

For every 1D effective subshift S, the 2D subshift :

$$S^{\mathbb{Z}} = \{y | \exists x \in S, \forall i, j, y_{ij} = x_i\}$$

$$S^{\mathbb{Z}} = \{y | \text{all lines are equal to the same } x \in S\}$$

is sofic.

Corollary

Theorem (Hochman-Meyerovitch [HM10])

For every right recursively enumerable real λ, there exists a 2D SFT of entropy λ.

Proof of the corollary

- Let λ be right-r.e.,
- Let $S_\lambda \subseteq \{0,1\}^{\mathbb{Z}}$ that forbids all words w so that the density of 1 in w is greater than λ : ($|w|_1 > \lfloor |w|\lambda \rfloor + 1$)
- S_λ is clearly effective.

In every infinite word of S_λ, the upper density of 1 is less than λ, and there are words where it is exactly λ (take a Sturmian word of slope λ)

From 1D to 2D

- Use Aubrun-Sablik to obtain a 2D SFT S'_λ that factors onto $S_\lambda^{\mathbb{Z}}$
- Look carefully at the construction, and see that S'_λ is *of zero entropy*

Now we replace every symbol x that maps into 1 by *two* different symbols x_1, x_2. Let's call X_λ this new SFT.

End of the proof

Let p_n be the number of patterns of size n in X_λ

$$p_n \leq p'_n 2^{\lambda n^2}$$

where p'_n is the number of patterns of size n in S'_λ. (There are at most λn^2 positions where we have to choose between x_1 and x_2)

$$p_n \geq 2^{\lambda n^2}$$

(If we start from a Sturmian word of density λ, we have at least λn^2 positions where we have a choice to make.)

$$\lim \frac{\log p_n}{n^2} = \lambda$$

($\lim \frac{\log p'_n}{n^2} = 0$ because S'_λ is of zero entropy).

Bibliography

Nathalie Aubrun and Mathieu Sablik, *Simulation of effective subshifts by two-dimensional SFT and a generalization*, preprint.

Bruno Durand, Andrei Romashchenko, and Alexander Shen, *Effective Closed Subshifts in 1D Can Be Implemented in 2D*, Fields of Logic and Computation, Lecture Notes in Computer Science, no. 6300, Springer, 2010, pp. 208–226.

Michael Hochman and Tom Meyerovitch, *A characterization of the entropies of multidimensional shifts of finite type*, Annals of Mathematics **171** (2010), no. 3, 2011–2038.

Shahar Mozes, *Tilings, substitutions systems and dynamical systems generated by them*, J. d'Analyse Math. **53** (1989), 139–186.