Reducibilities relations with applications to symbolic dynamics

Part III: Higman theorems

E. Jeandel

LORIA (Nancy, France)

Aubrun-Sablik look suspiciously like an old theorem from Higman on finitely generated groups

This analogy can be pushed further to obtain more results.

Warning: only an analogy in this talk! First part is informal!!

Finitely generated groups

- By $\mathfrak{G}_n$ we denote the "set" of all groups with generators $g_1 \dots g_n$.
- The "largest" group in $\mathfrak{G}_n$ is $\mathbb{F}_n$, the free group on n generators.
- Every other group with n generators is a quotient of $\mathbb{F}_n$: $G = F_n/N$ where N is a normal subgroup of F_n.
- A presentation of G is a subset R of N that generates N (as a normal subgroup). We write

$$G = \langle g_1, g_2 \dots g_n | R \rangle$$

- G is somehow the largest group generated by $g_1 \dots g_n$ in which all relations of R hold.

Finitely generated groups

Let $G = \left\langle a, b | aba^{-1}b^{-1} \right\rangle$

- G is the largest group in which $aba^{-1}b^{-1} = 1$, i.e. $ab = ba$.
- $G \simeq \mathbb{Z}^2$.

A group G is finitely presented if it can be given by a finite set R.
A group G is recursively presented if it can be given by a recursively enumerable set R.

f.g. groups vs subshifts

The construction of groups with presentations is the same as the construction of subshifts.

- Identify a group with the set of all identities in the group
- Identify a subshift with its forbidden language

In both cases, we start from a set of relations, and we see all relations we can obtain as consequences of these relations.

Semigroups with zero

Let S be the subshift that forbids ab and bba.
We can write

$$S = \langle a, b | ab = 0, bba = 0 \rangle$$

Then the set of all words of value 0 are (not) exactly the forbidden language of S!

The analogy

Groups	Shifts
Groups with n generators	Subshifts of $\{1, 2, \ldots, n\}^{\mathbb{Z}}$
Free group on n generators	Full shift on n letters
Finitely presented group	SFT
Recursively presented group	Effectively closed shift
$G_1 \subseteq G_2$	$L(S_1) = L(S_2) \cap A^\star$

Let's look at theorems on groups using this correspondence!

Plan

1 First Higman Theorem

2 Second Higman Theorem

3 Third Higman Theorem

4 Conclusion

First Higman Theorem

Theorem (Higman 1961 [Hig61])

G is recursively presented iff there exists a finitely presented group H s.t. $G \subseteq H$.

First Higman Theorem for subshifts

Theorem

$S \subseteq A^{\mathbb{Z}}$ is effectively closed iff there exists a SFT $S_2 \subseteq B^{\mathbb{Z}^2}$ s.t. $L(S) = L(S_2) \cap A^\star$

Note: $L(S_2)$ is a language of bidimensional words: we extract from it the one-dimensional words on the alphabet A.

Proof

- If S_2 is a SFT:
  - $L(S_2)$ is corecursively enumerable
  - $L(S_2) \cap A^\star$ is corecursively enumerable
  - $L(S)$ is corecursively enumerable
  - S is effectively closed.
- Suppose $S \subseteq A^{\mathbb{Z}}$ is effectively closed.
  - Use Aubrun-Sablik to obtain a $\mathbb{Z}^2$ SFT X over the alphabet $A \times B$ s.t. $S^{\mathbb{Z}} = \pi(X)$ where $\pi : (a, b) \to a$ is the canonical projection
  - Rewrite X as a $\mathbb{Z}^2$ SFT with every other row in A and every other row in B

Note: the result is apriori weaker than Aubrun-Sablik.

Plan

1. First Higman Theorem
2. Second Higman Theorem
3. Third Higman Theorem
4. Conclusion

Second Higman Theorem

Theorem (Boone-Higman-Thompson 1974-1980 [BH74, Tho80])

G has a recursive word problem iff there exists f.g. H_1, H_2 s.t. $G \subseteq H_1 \subseteq H_2$ with H_1 simple and H_2 finitely presented.

G has a recursive word problem iff $G \subseteq H$ for some recursively presented, simple group H.

Second Higman Theorem for subshifts

Theorem (J.-Vanier 2017)

Let $S \subseteq A^{\mathbb{Z}}$ be a subshift.
Then $L(S)$ is recursive iff there exists an effectively closed minimal 2D-subshift S_1 s.t. $L(S) = L(S_1) \cap A^\star$

Corollary (Durand-Romashchenko 2017)

Let $S \subseteq A^{\mathbb{Z}}$ be a subshift.
Then $L(S)$ is recursive iff there exists a minimal 3D-SFT S_2 s.t. $L(S) = L(S_2) \cap A^\star$.

Minimality plays the role of Simplicity.
Why is the corollary not immediate ?

Proof

We prove already that if X is effectively closed and minimal, then $L(X)$ is recursive. Therefore $L(S) = L(X) \cap A^\star$ is also recursive.
For the converse, we first ask:

What does it mean for a minimal X to be effectively closed ?

Minimal subshifts

Theorem

X minimal is effectively closed iff it contains a recursive point and its quasiperiodicity function is recursive.

$f(n) =$least m s.t. every $n \times n$ pattern in $L(X)$ appear in all $m \times m$ patterns in $L(X)$.

- Suppose X is effectively closed and minimal. Then $L(X)$ is recursive. Therefore:
  - X has a recursive point (actually a dense subset of recursive points)
  - f is computable by definition.
- Suppose $x \in X$ is computable and f is computable
  - Then $D(X)$ is exactly the set of $n \times n$ patterns that do not appear in the square of size $f(n) \times f(n)$ around the origin of x.

Proof

Let $L(S)$ be computable. How to do a two-dimensional minimal subshift where every word of $L(S)$ appear ?

Proof

00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#

Proof

00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#

Proof

00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#

Proof

00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#

Proof

00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#

Proof

00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#

Idea

Starting from a subshift S with computable language, we create a configuration x :

- Every row has a level: The level of the row $j \times 2^n$, with j odd, is n.
- Rows of level n contains periodically all patterns of $L(S)$ of size $n+1$, separated by a $\#$ symbol.
- Row of level 0 contains any infinite word

This configuration x is computable (provided some care is done for level 0) and the subshift generated by x is minimal.

Problem: the quasiperiodicity function is not computable.

Realization

- Row of level n contains periodically all possible pairs of patterns $u\#v$ for $u, v \in L(S)$ of size p_n
  - Ensures that patterns of small size that appear in row of large level also appear in rows of small level.
- $p_n - 1$ is precisely the size of the period of the rows of level $n - 1$.
  - The $\#$ symbol will be *synchronized*.

No pictures!

Going further

$L(S) = L(S_2) \cap A^\star$ is not very nice, what about a factor map ?

Theorem (Tentative Theorem)

Let $S \subseteq A^{\mathbb{Z}}$ be a subshift.
Then $L(S)$ is recursive iff it is a factor of a minimal SFT.

Going further

This does not work:

- If S is a factor of a minimal system, then S has dense minimal points.
  - Not every recursive subshift can be obtained this way

Going further

If S is a recursive 1D subshift, then

- $S \cap A^{\mathbb{Z}}$ is not always recursive.
- $L(S) \cap A^{\star}$ is recursive.

The theorem is about finite words, not infinite words.

Corollary

Let $S \subseteq A^{\mathbb{Z}}$ be a subshift.
Then S is effectively closed iff there exists a minimal 3D-SFT S_2 s.t.
$S = \mathrm{rows}(S_2) \cap A^{\mathbb{Z}}$

Plan

1. First Higman Theorem
2. Second Higman Theorem
3. Third Higman Theorem
4. Conclusion

Finitely presented in

Definition

G is finitely presented in *H* if *G* is obtained from *H* by adding finitely many generators and finitely many relations.

(The definition usually also requires that $H \subseteq G$)

Third Higman Theorem

Theorem (Ziegler, CF Miller III)

G is a subgroup of a finitely presented group in H iff $\mathrm{WP}(G) \leq_e \mathrm{WP}(H)$

Recall what $\leq_e$ means:

- From an enumeration of the word problem of H, I can enumerate the word problem of G.
- From an enumeration of the words w s.t. $w = 1$ in H, I can enumerate all words w s.t. $w = 1$ in G.

What is the equivalent for subshifts ?

SFT in

Definition

S is a SFT in T if S is obtained from T by adding finitely many letters and finitely many forbidden words.
More precisely $S = X_{D(T) \cup F}$ for some finite set F.

We also add the possibilty for S to be in a higher dimension than T.

Example(s)

Let T be a subshift over the alphabet A in dimension 1

- Let S be the subshift over the alphabet A in dimension 2 with the same forbidden patterns. What does S look like ?
- We forbid different letters in vertically adjacents positions. What does S look like now ?
- Let S be the subshift over the alphabet $A \cup \{\#\}$ in dimension 1 with the same forbidden patterns. What does S look like ?

Third Higman theorem

Theorem (J.-Vanier 2017)

Let S, T be two subshifts over disjoint alphabets A and B.
Then $D(S) \leq_e D(T)$ iff $L(S) = L(S_1) \cap A^\star$ for some subshift S_1 that is a SFT over T.

Corollary

Start from a subshift T and apply some of the following operations:

- *Change the alphabet (add letters, or rename them)*
- *Add some forbidden pattern*
- *Change the dimension (up or down)*

The subshifts we obtain are exactly all subshifts S s.t. $D(S) \leq_e D(T)$.

Warning

This theorem is very similar to a theorem of Aubrun and Sablik 2009 [AS09]

- Same operations (plus product and factor), except we cannot add letters
- Different conclusion: $D(S) \leq_s D(T)$, where s is strong reducibility

However, the proof in the article are *completely wrong* in both directions.

Idea of the proof

One direction is OK.
Other direction: Start from $D(S) \leq_e D(T)$. Suppose S and T are one-dimensional to simplify, and S is over $\{0, 1\}$.

- This is a statement about finite words rather than infinite words
  - Extract $L(T)$ from T
  - Deduce the words of $L(S)$ from $L(T)$
  - Recombine $L(S)$ into S.

Two steps

First part: Starting from T, we build a 1D subshift S_1 s.t. configurations of S_1 :

- Either contain at most one letter $\#$
- Or are periodic of the form $\# w_1 \# w_2 \dots w_{2^n}$ where each w_i is in $L(S)$ and of length n.

Second part: Starting from S_1, we build S.
How to do the second part ?

Proof

00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#0101#1000#1001#1010#0000#0001#0010#0100#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#000#001#010#100#101#
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#00#01#10#0
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#
00000#00001#00010#00100#00101#01000#01001#01010#10000#10001#10010#10100#10101#00000#00001#00010#0010
0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#0#1#

Same construction, different semantics

- Row of level *n* contains (periodically) 2^n words of size *n*.
- Each factor of a word in level $i \in [1, +\infty]$ appears in level *j* for $j < i$.

The subshift generated by these configurations is effectively closed. (Why ?)

- Suppose now that each row is in S_1.
- Then necessarily the only rows with no $\#$ symbols are in *S*.
- We can extract these rows with down + Forbidden patterns.

First Part

First part: Starting from T, we build a 1D subshift S_1 s.t. configurations of S_1 :

- Either contain at most one letter $\#$
- Or are periodic of the form $\#w_1\#w_2\dots w_{2^n}$ where each w_i is in $L(S)$ and of length n.

We know that $D(S) \leq_e D(T)$. What does it mean for $L(S)$ relatively to $L(T)$?

Suppose that $D(S) \leq_e D(T)$. Then there exists a recursive function F s.t. on input p and u outputs a finite set $F(p, u)$ of words s.t.

$$u \in D(S) \iff \exists p, F(p, u) \subseteq D(T)$$

$$u \in L(S) \iff \forall p, F(p, u) \cap L(T) \neq \emptyset$$

First part

Build a 2D subshift s.t. rows are either over the alphabet $\{0, 1, \#\}$ or the alphabet $A \cup \{\sharp\}$ (A is the alphabet of T) s.t:

- There is at most one row with elements in $\{0, 1, \#\}$
- If this row contains two $\#$ symbols, then
  - it is periodic of the form $\#w_1\#w_2 \dots w_{2^n}$ where each w_i is of length n.
  - The p-th row above this row contains a periodic word of the form $\sharp u_1 \sharp u_2 \dots \sharp u_{2^n}$ where $u_i \in F(p, w_i)$

This subshift is clearly effectively closed (why ?)

- Now suppose that all rows over $A \cup \{\sharp\}$ actually contains $T_\sharp$, the shift T with the additional $\sharp$ letter.
- Then the row over $\{0, 1, \#\}$ should be in $L(S)$!

S_1 can therefore be obtained by down + forbidden patterns.

Plan

1. First Higman Theorem
2. Second Higman Theorem
3. Third Higman Theorem
4. Conclusion

Generalizations

Can we obtain other new theorems by this correspondence?

How(What is the best way) to formalize this correspondence?

The fourth Higman theorem

What about the fourth Higman theorem ?

Realization

Definition

Let S be a subshift over an alphabet A Let u_i, v_i be a finite set of patterns over the alphabet $A \cup X$.
A subshift T over an alphabet $A \cup B$ realizes (S, u_i, v_i) if, up to a renaming of the letters of X into the letters of B:

- $S = T \cap A^{\mathbb{Z}}$.
- All patterns u_i are forbidden in T, all patterns v_i appear in T.

A tuple (u_i, v_i, S) is realizable if there exists a subshift that realizes it.

Examples

Let S be the subshift that forbids 01^p0 for p prime.

- Let $u_1 = 0x0$ and $v_1 = 1x1$.

Then the subshift T over $\{0, 1, 2\}$ that forbids 01^p0 and 020 realizes (S, u_1, v_1).
Actually, any subshift $T' \subseteq T$ also realizes (S, u_1, v_1) as long as it contains S and the pattern 121 is allowed.

e.c. Shift

Definition

A subshift $T \subseteq \Delta^{\mathbb{Z}}$ is existentially closed if every realizable tuple is realized:

- For every subshift $S = T \cap A^{\mathbb{Z}}$ for some finite alphabet A, and every realizable tuple (S, u_i, v_i), the subshift T realizes (S, u_i, v_i)

Δ is an infinite alphabet!

Theorem (Tentative theorem)

S is recursive iff it is included in any existentially closed subshift.

Bibliography I

- Nathalie Aubrun and Mathieu Sablik, *An order on sets of tilings corresponding to an order on languages*, 26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings, 2009, pp. 99–110.
- William W. Boone and Graham Higman, *An algebraic characterization of groups with soluble word problem*, Journal of the Australian Mathematical Society **18** (1974), no. 1, 41–53.
- Graham Higman, *Subgroups of Finitely Presented Groups*, Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences **262** (1961), no. 1311, 455–475.

Bibliography II

Richard J. Thompson, *Embeddings into finitely generated simple groups which preserve the word problem*, Word Problems II (Sergei I. Adian, William W. Boone, and Graham Higman, eds.), Studies in Logic and the Foundations of Mathematics, vol. 95, North Holland, 1980, pp. 401–441.