Communication Complexity for Multidimensional subshifts

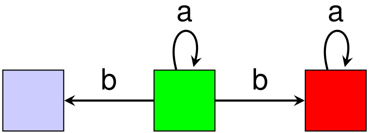
Towards Characterizing Soficness

E. Jeandel

LORIA (Nancy, France)

Plan

Sofic shifts in 1D



$$L = \{\dots aaaa \dots, \dots aabaa \dots\}$$

SFTs and Sofic Shifts

Definition

A subset $S \subseteq \Sigma^{\mathbb{Z}}$ is a sofic shift iff it is the set of biinfinite words corresponding to a domino system

Definition

A subset $S \subseteq \Sigma^{\mathbb{Z}}$ is a sofic shift iff it is the set of biinfinite paths on some finite graph.

- S is a “regular language” of infinite words.
- Can be described by a finite automaton.
- Sofic shifts are closed under union, intersection, etc and we can prove it with finite automata.

Other definition

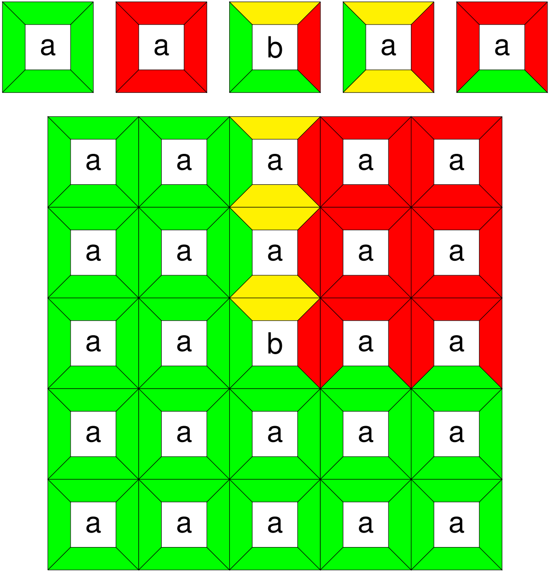
Definition

A set S of biinfinite words is a subshift if it can be defined by a set of forbidden words $\mathcal{F}$.

- $\mathcal{F}$ finite : S is said to be of finite type (SFT)
- $\mathcal{F}$ regular : S is sofic

Note : dominoes represent of shift of finite type (SFT). In fact sofic shifts can be defined as “projections” of SFTs.

Sofic shifts in 2D



Sofic shifts in 2D

- No notions of deterministic automata
- No characterizations of regular languages
- No algorithm to decide if a regular language is empty
  - From automata to Turing machines

Nevertheless, we would like to have criteria to prove something is (not) sofic.

How to prove soficness

How to prove something is sofic

- Usually by building the domino system.
- Ex : The set S of configurations over $\{0, 1\}$ where every finite connected component of 1 is of even size is sofic (Cassaigne).

Very few general statements.

- Every "substitutive" shift is sofic (reference depends on how to interpret the quotes)
- Everything expressed by a $\exists X \forall y$ formula is sofic (Jeandel-Theyssier)
- Aubrun-Sablik

How to disprove soficness

Usually by proving that the set S does not have a property shared by all sofic shifts.

- A sofic shift has a right-enumerable entropy (Hochman-Meyerovitch. . .)
- A sofic shift contains a configuration of “low” Kolmogorov complexity.

Rationale here

- 2D sofic shifts are hard to understand
- 1D sofic shifts are easy to understand

Look at 1D shifts inside 2D shifts.

First approach

Let S be a language of pictures for which all lines are identical. Let S_1 be the corresponding unidimensional language.

- When is S sofic ?

Theorem (Durand-Romashchenko-Shen, Aubrun-Sablik 2010)

S is sofic exactly when S_1 is effective (can be given by a computable family $\mathcal{F}$ of forbidden words)

From 2D to 1D : second approach

Given a 1D language S_1 we look at the set of all pictures S where every line is in S_1.

- No correlation between the different lines

We know of no example where S is sofic but S_1 is not.

Conjecture : S is sofic iff S_1 is.

In this talk : some advances towards this problem.

Plan

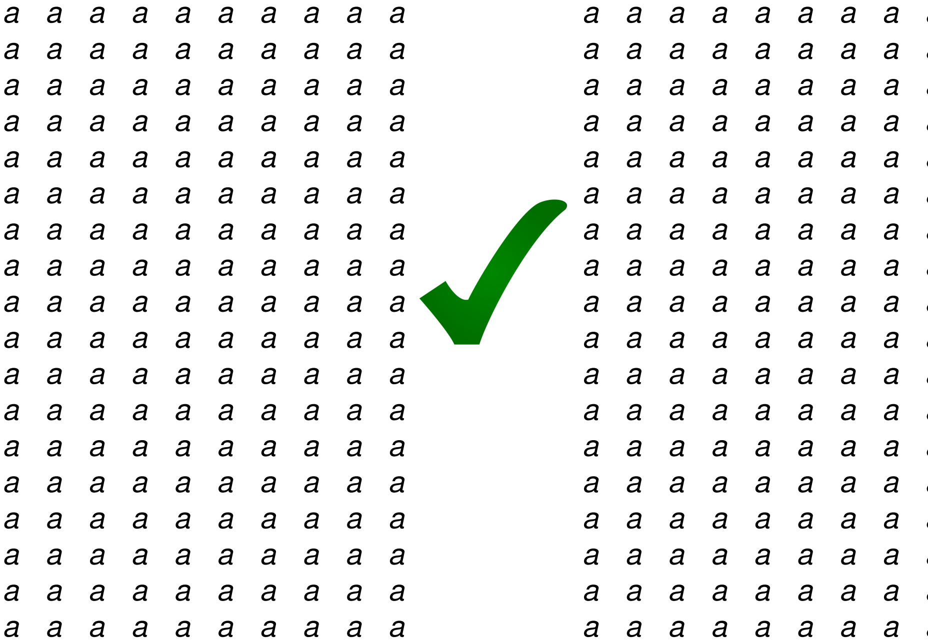
The idea

- Divide the plane into two halfs.
- Give the first half to Almighty Alice, the second one to Almighty Bob.

How much information should they exchange to decide whether they would obtain a valid picture by putting the two halfs together ?

a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a

a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a
a a a a a a a a



a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*

a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*b*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*

a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a

✓

a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	b	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a

a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*b*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*

a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*b*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*
a	*a*	*a*	*a*	*a*	*a*	*a*	*a*

a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	b	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a

a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	b	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a

Protocol

If S is sofic, there is a protocol that exchanges few bits :

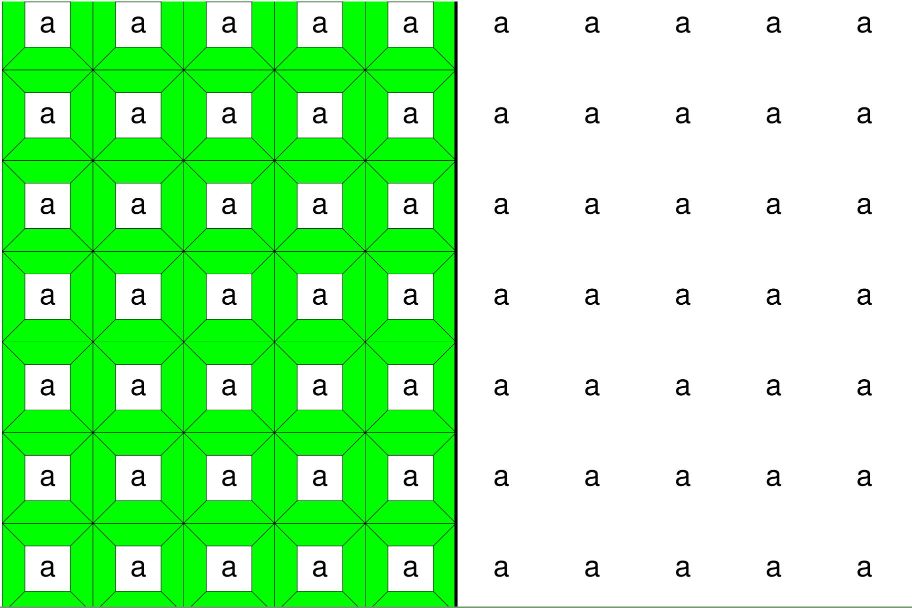
- Alice decides on how to tile its part of the plane.
- Alice sends the boundary to Bob
- Bob checks if it can tile its part of the plane with the same boundary as Alice.

If Alice makes the good choice, this protocol will succeed (non deterministic protocol).

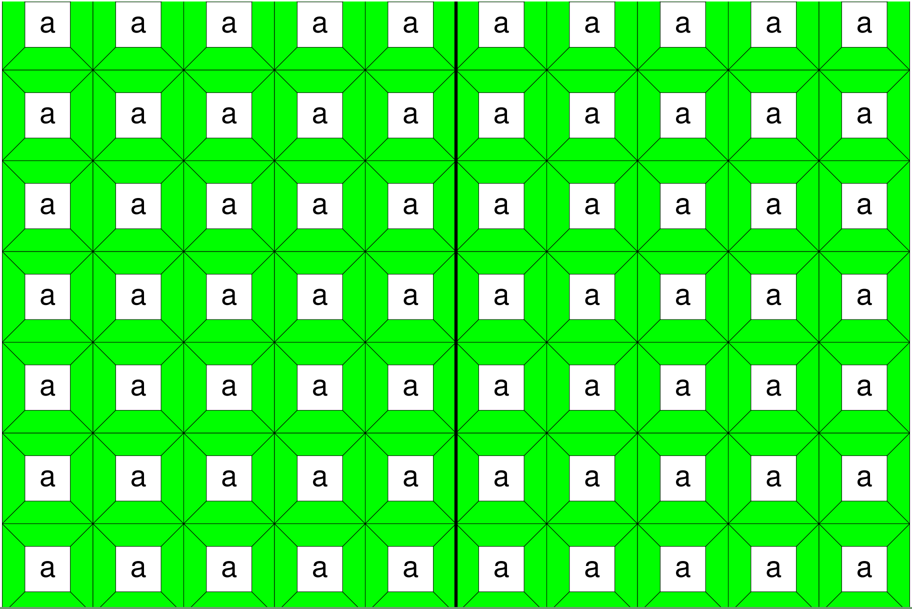
First example

a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a

First example



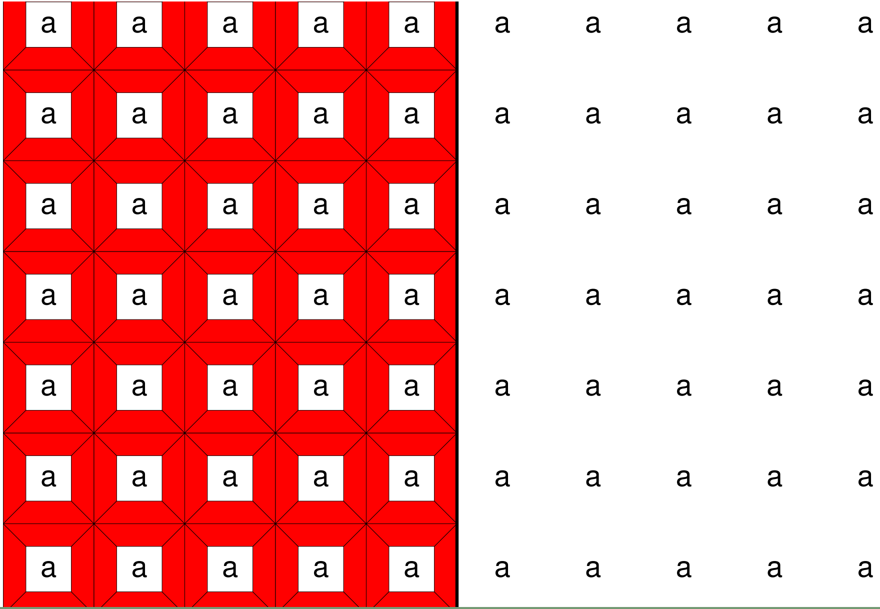
First example



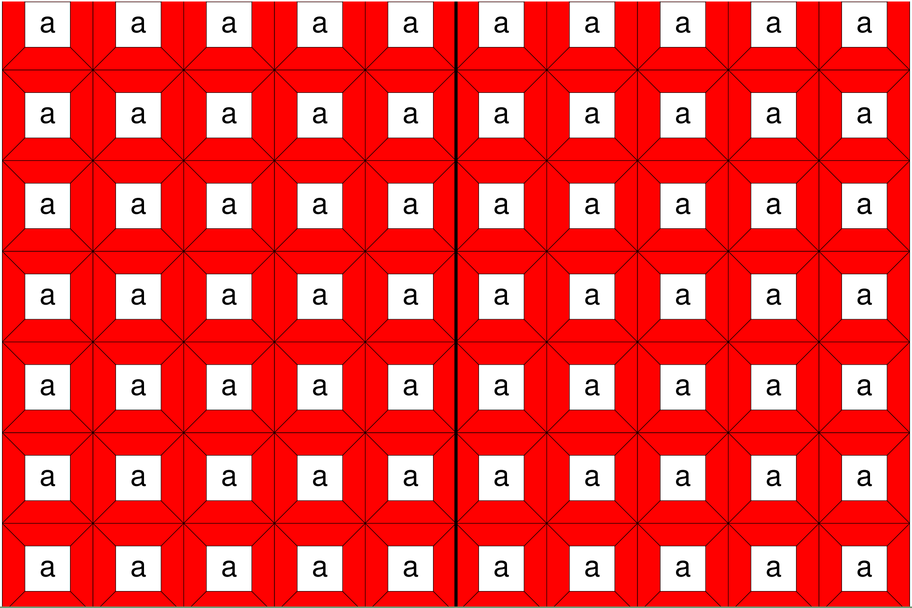
First example

a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a

First example



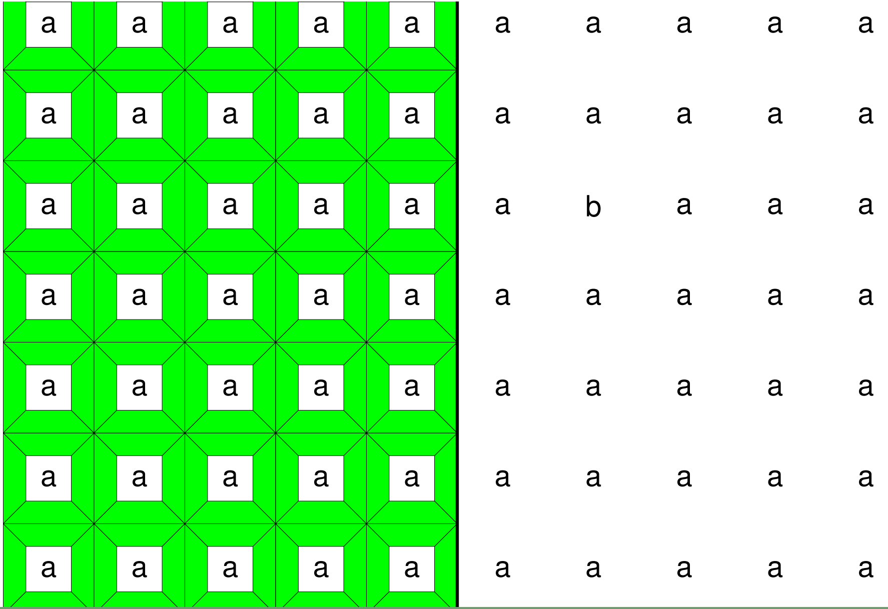
First example



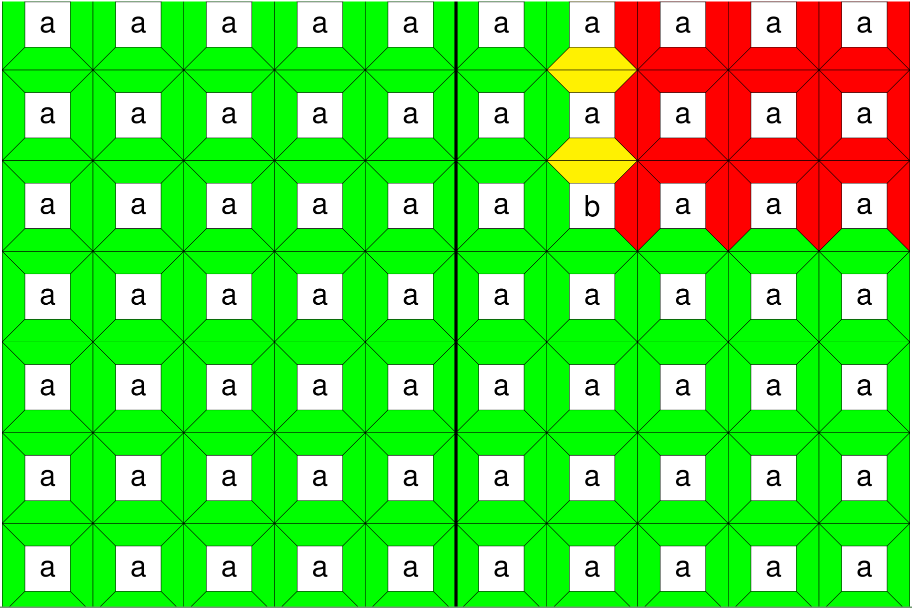
Second example

a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	b	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a

Second example



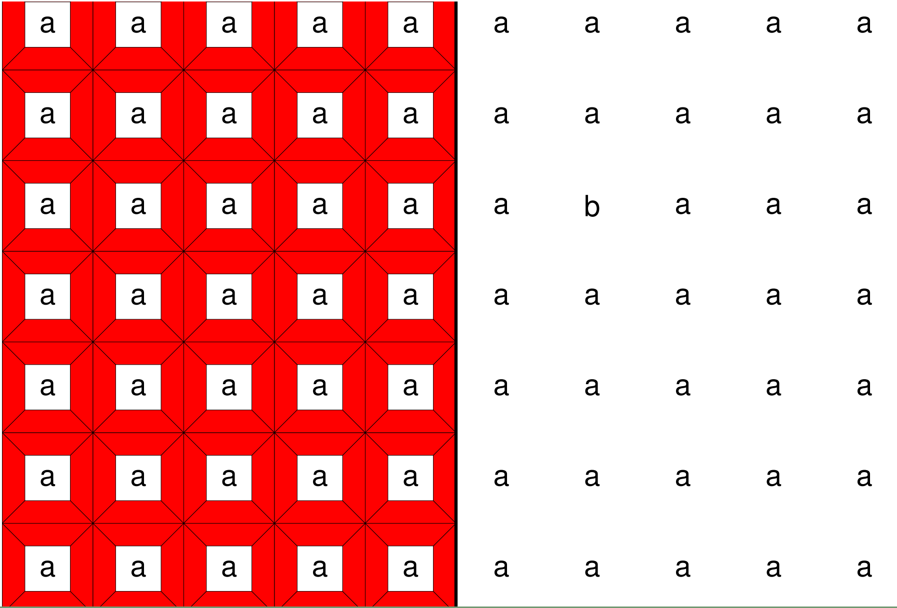
Second example



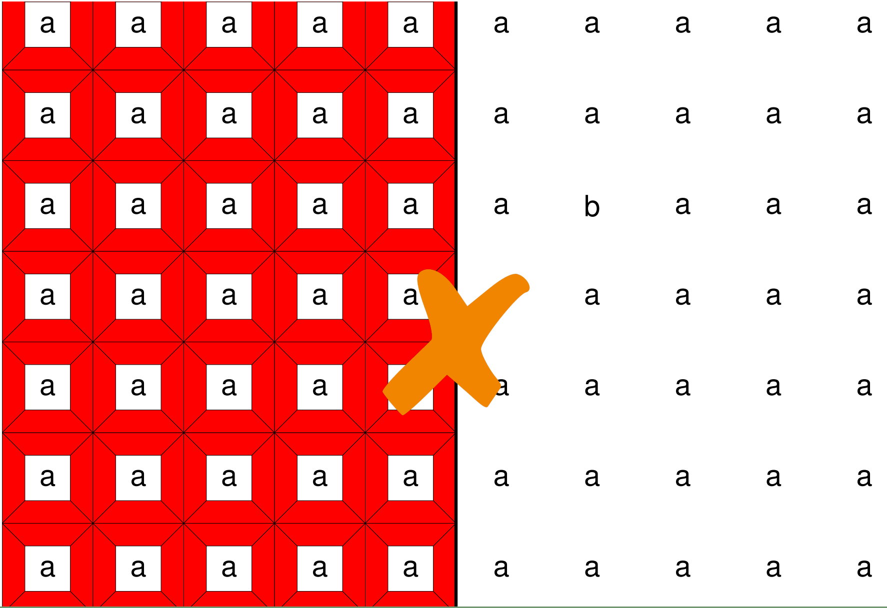
Second example

a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	b	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a

Second example



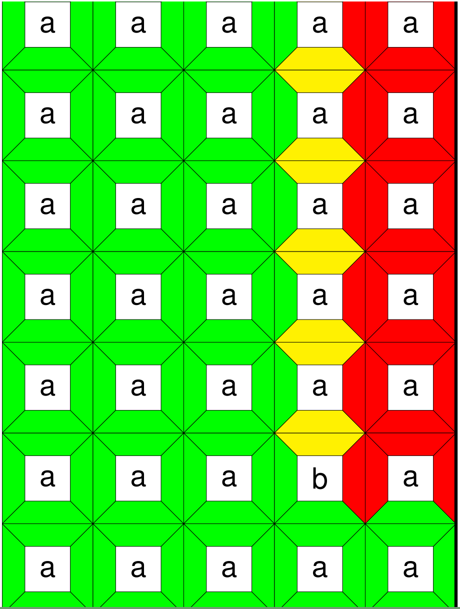
Second example



Third example

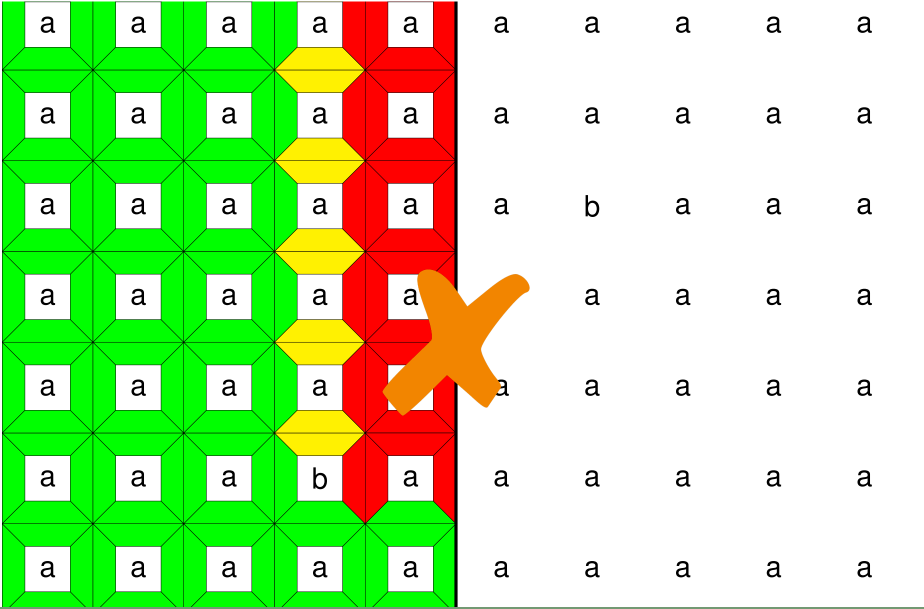
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	b	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	b	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a

Third example



a	a	a	a	a
a	a	a	a	a
a	b	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a

Third example



Communication Complexity

We now give formal definitions.

- We also symmetrize the protocol. Both Alice and Bob are given some boundary x, and they each verify that they can tile their half of the plane.
- To simplify things, we will only give to Alice and Bob the first n columns of their half, and not the whole half.
- This means that Alice and Bob both have an element in a one-dimensional (vertical) subshift.

Communication Complexity

Definition

Let $S \subset A \times B$ be a subshift (A and B are also subshifts)
A *protocol* for S is three subshifts X, P_A, P_B so that :

$$(a, b) \in S \iff \exists x \in X, (a, x) \in P_A \wedge (b, x) \in P_B$$

- Alice has $a \in A$, obtains x and tests whether $(a, x) \in P_A$
- Bob has $b \in B$, obtains x and tests whether $(b, x) \in P_B$

Communication Complexity

Definition

The communication complexity $CC(S)$ of a subshift S is the infimum of $h(X)$ for a protocol (X, P_A, P_B) for S.

$h(X)$ is the entropy of X. $h(\{0, \dots k\}^{\mathbb{Z}}) = \log k$.

Some trivial facts

- $CC(S) \leq h(A)$ (We can always send Alice's input to Bob)
- $CC(A \times B) = 0$ (Nothing to transmit)

Let S_1 be any subshift and $EQ = \{(a, a) | a \in S_1\}$

$$CC(EQ) = h(S_1)$$

Proof for EQ

Let S_1 be any subshift and $EQ = \{(a, a) | a \in S_1\}$

$$CC(EQ) = h(S_1)$$

- $CC(EQ) \leq h(S_1)$ is clear.

Let (X, P_A, P_B) be a protocol for EQ.

- To each element $x \in X$ corresponds at most one element of S_1, wlog exactly one.
- We can prove that the map $X \to S_1$ is then a factor map
- Hence $h(X) \geq h(S_1)$.

Strange example

$$EQ_{:/} = (\{0,1\} \times \{0,1\})^{\mathbb{Z}} \cup \{(0,0),(1,1)\}^{-\omega} 2 \{(0,0),(1,1)\}^{\omega}$$

If Alice and Bob both have a 2, they should have the same word.

$$CC(EQ_{:/}) = 0$$

Strange example

- If Alice has a 2, she sends all her information to Bob in a sparse way
  - Alice has
  $$\cdots 010010101021000 1010 \cdots$$
  - She sends
  $$\cdots 0\sharp\sharp\sharp\sharp 1\sharp\sharp\sharp 0\sharp\sharp 1\sharp 021\sharp 0\sharp\sharp 0\sharp\sharp\sharp 0\sharp\sharp\sharp\sharp 1 \cdots$$
- Otherwise she sends ${}^{\omega}\sharp^{\omega}$ (possibly with one $0/1$ symbol at some place)

Should this example be forbidden somehow ?

Plan

Multidimensional subshifts

Proposition

Let S be a two-dimensional subshift.
Let C_n be the shift of n consecutive columns of S.

$$S_{n,m} = \{(a,b) \in C_n \times C_m | ab \in C_{n+m}\}$$

If S is sofic, then $CC(S_{n,m}) = O(1)$.

- This is “tight”, in the sense that a similar proposition for 1D subshift characterize sofic subshifts.

Special case S is an SFT

Theorem

if S is a SFT, then $CC(S)$ is the infimum of $h(X)$ for finite type protocols (X, P_A, P_B of finite type)

Let (X, P_A, P_B) a protocol.
We can suppose that P_A and P_B are SFTs :

- Let P_A^n, P_B^n be upper approximations of P_A and P_B by forbidding only patterns of size n.
- We obtain a protocol for a upper approximation of S.
- As S is defined by finitely many forbidden patterns, for some n, we will obtain exactly S.

Losing only ϵ in entropy, we can suppose that X is sofic.

- $X' = \{x | \forall (a,b) \in A \times B, (a,x) \in P_A \wedge (b,x) \in P_B \implies (a,b) \in S\}$
- $X' \supset X$ is sofic, and defines the same set S.
- We can make X' closer to X in entropy while preserving soficness

We can assume X SFT by changing the protocol (every sofic shift is factor of a SFT of same entropy)

Some notes

- The theorem does not work for sofic shifts : The previous “bad” example was sofic, but the protocol is not sofic.
- The proof does not work in higher dimensions.

A corollary

Definition

Let Σ be a finite set, and $R \subseteq \Sigma \times \Sigma$
If we change subshift into finite set and $h(X)$ into $\log |X|$ into the previous definition, we obtain the communication complexity $N(R)$ of a relation.

Theorem

Let $A = B = \Sigma^{\mathbb{Z}}$ and $S = R^{\mathbb{Z}}$.
Then $CC(S) = N^{asymp}(R)$ where $N^{asymp}(R) = \lim_{n\to\infty} N(R^m)/m$

$N^{asymp}(R)$ is well studied in Communication Complexity.

The original question

Let's go back to the original question.
S_1 a 1D shift. S a 2D shift where all lines are in S.

Does S sofic implies S_1 sofic ?

What is C_n (the set of n columns of S) ?
By definition $C_n = L_n^{\mathbb{Z}}$, where L_n is the set of words of size n of S_1.

New Result

Theorem

Let $R_n = \{(x, y) \in L_n | xy \in L_{2n}\}$
Then $CC(S_{n,n}) \geq N(R_n) - \log\log L_n + O(1)$
In particular, if $N(R_n) - \log\log L_n \neq O(1)$*, then S is not sofic.*

Direct translation of a result about asymptotic communication complexity (Feder et al 91)

- If $N(R_n) > \log\log L_n + O(1)$, S is not sofic.
- If $N(R_n) = O(1)$, S_1 is sofic.
- It remains to fill the gap.

Implies the result by Pavlov that if S_1 has no synchronizing word, then S is not sofic.

Plan

Open questions

- Find more properties of $CC(S)$
- Is $CC(S)$ always achieved by some protocol ?
- Link with conditional entropy ?
- Look at the case where A and B are general zero-dimensional systems (we give the whole half to Alice and Bob)
- Translate lower bounds from finite CC into results on shifts.

An example

Theorem

$N(R) = \max_\mu \min_{R_1 \times R_2 \subseteq R} - \log \mu(R_1 \times R_2)$