Towards a classification of transitivity classes for Hom shifts

S. Gangloff*, joint work with B. Hellouin** and P. Oprocha*

* AGH, Faculty of Applied Mathematics, Kraków, ** Laboratoire de recherche en Informatique, Orsay.

sgangloff@agh.edu.pl; silvere.gangloff@gmx.com
Motivations
Bidimensional SFT: bidimensional dynamical system corresponding to the $\mathbb{Z}^2$-action of the shift.
Bidimensional SFT : bidimensional dynamical system corresponding to the $\mathbb{Z}^2$-action of the shift on a subset of $A^\mathbb{Z}^2$ defined by a finite set of forbidden patterns.
**Bidimensional SFT** : bidimensional dynamical system corresponding to the $\mathbb{Z}^2$-action of the shift on a subset of $\mathcal{A}^{\mathbb{Z}^2}$ defined by a finite set of *forbidden* patterns.

Ex : **Hard square shift** ; $\mathcal{A} = \{0, 1\}$
Bidimensional SFT: bidimensional dynamical system corresponding to the $\mathbb{Z}^2$-action of the shift on a subset of $\mathcal{A}^{\mathbb{Z}^2}$ defined by a finite set of forbidden patterns.

Ex: Hard square shift; $\mathcal{A} = \{0, 1\}$

Forbidden patterns $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ et $\begin{pmatrix} 1 ; 1 \end{pmatrix}$. 
Bidimensional SFT: bidimensional dynamical system corresponding to the \( \mathbb{Z}^2 \)-action of the shift on a subset of \( \mathcal{A}^{\mathbb{Z}^2} \) defined by a finite set of forbidden patterns.

Ex: Hard square shift; \( \mathcal{A} = \{0, 1\} \)

Forbidden patterns \begin{pmatrix} 1 \\ 1 \end{pmatrix} and \begin{pmatrix} 1; 1 \end{pmatrix}.

\[
\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]
Bidimensional SFT : bidimensional dynamical system corresponding to the $\mathbb{Z}^2$-action of the shift on a subset of $A^{\mathbb{Z}^2}$ defined by a finite set of forbidden patterns.

Ex : Hard square shift ; $A = \{0, 1\}$

Forbidden patterns $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ et $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$. 

\[
\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]
**Bidimensional SFT**: bidimensional dynamical system corresponding to the $\mathbb{Z}^2$-action of the shift on a subset of $\mathcal{A}^{\mathbb{Z}^2}$ defined by a finite set of *forbidden* patterns.

Ex: **Hard square shift**; $\mathcal{A} = \{0, 1\}$

Forbidden patterns: $\begin{array}{c}1 \\ 1 \end{array}$ et $\begin{array}{c}1 \end{array}$; $\begin{array}{c}1 \end{array}$.

```
0 0 0 0 0 0
- - - - - -
0 0 0 1 0 0
- - - - - -
0 0 0 0 1 0
- - - - - -
1 0 1 0 0 0
- - - - - -
0 0 0 0 0 0
```
Bidimensional SFT: bidimensional dynamical system corresponding to the $\mathbb{Z}^2$-action of the shift on a subset of $\mathcal{A}^{\mathbb{Z}^2}$ defined by a finite set of forbidden patterns.

Ex: Hard square shift; $\mathcal{A} = \{0, 1\}$

Forbidden patterns $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ et $\begin{bmatrix} 1 ; 1 \end{bmatrix}$.

```
0 0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 1 1   \text{oops}
1 0 1 0 0
0 0 0 0 0
```
**Bidimensional SFT**: bidimensional dynamical system corresponding to the $\mathbb{Z}^2$-action of the shift on a subset of $\mathcal{A}^{\mathbb{Z}^2}$ defined by a finite set of forbidden patterns.

*Ex*: **Hard square shift**; $\mathcal{A} = \{0, 1\}$

Forbidden patterns $\begin{array}{c} 1 \\ 1 \end{array}$ et $\begin{array}{c} 1 \ 1 \end{array}$. 

```
0 0 0 0 0 0
0 0 0 1 0
0 0 0 0 1
1 0 1 0 0
0 0 0 0 0
```
Entropy and computability:

Let $X$ be a bidimensional SFT.
Entropy and computability:

Let $X$ be a bidimensional SFT.

**Entropy**: $\inf_n \frac{\log(N_n(X))}{n^2}$, where $N_n(X)$ is the number of $n$-square which appear in at least one element of $X$. 

**Computability**: $x \in \mathbb{R}$ is computable when there is an algorithm which approximates $x$ with elements of $\mathbb{Q}$ with arbitrary precision.
Entropy and computability:

Let $X$ be a bidimensional SFT.

**Entropy**: $\inf_n \frac{\log(N_n(X))}{n^2}$, where $N_n(X)$ is the number of $n$-square which appear in at least one element of $X$.

**Computability**: $x \in \mathbb{R}$ is computable when there is an algorithm which approximates $x$ with elements of $\mathbb{Q}$ with arbitrary precision.
A computational 'transition':

$f$-Block gluing:

Worldmap:

No man’s land

Algorithmic computability domain

Liminal area

Swamp of undecidability

[G., Hellouin] $o(\log(n))$

O(n) [G., Sablik]

[G., Hochman, Meyerovitch]
The question of intermediate gap functions

**Question** [G., Sablik, also related by M. Hochman]: does there exist some $f$-block gluing bidimensional SFT with undecidable language and $\log(n) = o(f(n))$ and $f(n) = o(n)$?
The question of intermediate gap functions

Question[G., Sablik, also related by M. Hochman]: does there exist some \( f \)-block gluing bidimensional SFT with undecidable language and \( \log(n) = o(f(n)) \) and \( f(n) = o(n) \)?:

Natural idea for \( f(n) = \sqrt{n} \) (fails):
The question of intermediate gap functions

**Question**[G., Sablik, also related by M. Hochman]: does there exist some \( f \)-block gluing bidimensional SFT with undecidable language and \( \log(n) = o(f(n)) \) and \( f(n) = o(n) \)?

Natural idea for \( f(n) = \sqrt{n} \) (fails):
The question of intermediate gap functions

Question [G., Sablik, also related by M. Hochman]: does there exist some $f$-block gluing bidimensional SFT with undecidable language and $\log(n) = o(f(n))$ and $f(n) = o(n)$?

Natural idea for $f(n) = \sqrt{n}$ (fails):

![Diagram of SFT pattern]
The question of intermediate gap functions

Question [G., Sablik, also related by M. Hochman]: does there exist some $f$-block gluing bidimensional SFT with undecidable language and $\log(n) = o(f(n))$ and $f(n) = o(n)$?

Natural idea for $f(n) = \sqrt{n}$ (fails):
The question of intermediate gap functions

**Question** [G., Sablik, also related by M. Hochman]: does there exist some $f$-block gluing bidimensional SFT with undecidable language and $\log(n) = o(f(n))$ and $f(n) = o(n)$?

Natural idea for $f(n) = \sqrt{n}$ (fails):

![Diagram of a grid with shaded squares and circles representing a possible SFT configuration.]
The question of intermediate gap functions

**Question** [G., Sablik, also related by M. Hochman] : does there exist some $f$-block gluing bidimensional SFT with undecidable language and $\log(n) = o(f(n))$ and $f(n) = o(n)$?

Natural idea for $f(n) = \sqrt{n}$ (fails) :
The question of intermediate gap functions

**Question** [G., Sablik, also related by M. Hochman]: does there exist some $f$-block gluing bidimensional SFT with undecidable language and $\log(n) = o(f(n))$ and $f(n) = o(n)$?

Natural idea for $f(n) = \sqrt{n}$ (fails):

**Problem**: it is actually linear block gluing.
Homshifts
Homshift: SFT $X_G$ whose forbidden patterns are:

```
  a
 b,  a b,
```

where $(a, b)$ not an edge in $G$ (non-oriented simple graph).
**Homshift**: SFT $X_G$ whose forbidden patterns are:

- $a$
- $b$
- $ab$,

where $(a, b)$ not an edge in $G$ (non-oriented simple graph).

The hard square shift is a homshift:

![Diagram of a hard square shift with states 0 and 1 connected by an arrow]
**Homshift**: SFT $X_G$ whose forbidden patterns are:

\[
\begin{array}{cc}
 a \\
 b, \\
 a b,
\end{array}
\]

where $(a, b)$ not an edge in $G$ (non-oriented simple graph).

**The hard square shift is a homshift**:

\[
\begin{array}{c}
 0 \\
 1
\end{array}
\]

**Interest**: symmetries break down undecidability phenomena; in general: the language is decidable, the entropy is computable (Friedland).
What are the possible gap functions for Hom shifts?
What are the possible gap functions for Hom shifts?

Simplifications:

1. Block gluing $\rightarrow$ Vertical transitivity.
2. Gap functions $\rightarrow$ Classes for the equivalence $f \sim g$ defined by for all $n$:
   
   $c + kf(n) \leq g(n) \leq c' + k'f(n)$. 
What are the possible gap functions for Hom shifts?

Simplifications:

1. Block gluing $\rightarrow$ Vertical transitivity.
What are the possible gap functions for Hom shifts?

Simplifications:

1. Block gluing $\rightarrow$ Vertical transitivity.

2. Gap functions $\rightarrow$ Classes for the equivalence $f \sim g$ defined by for all $n$:

$$c + kf(n) \leq g(n) \leq c' + k'f(n).$$
Expected result:

**Theorem**: *The transitivity classes for bidimensional Homshifts are Θ(1), Θ(log(n)) and Θ(n).*
Expected result:

**Theorem**: The transitivity classes for bidimensional Homshifts are $\Theta(1), \Theta(\log(n))$ and $\Theta(n)$.

Proven part: if not $\Theta(n)$ then $O(\log(n))$. 

Builds on tools developed by B. Marcus and N. Chandgotia.
Expected result:

**Theorem:** The transitivity classes for bidimensional Homshifts are $\Theta(1), \Theta(\log(n))$ and $\Theta(n)$.

Proven part: if not $\Theta(n)$ then $O(\log(n))$.

Builds on tools developed by B. Marcus and N. Chandgotia.
For c vertex, the **universal cover** \( \mathcal{U}_c(G) \) of \( G \) is the graph s.t. : i) vertices : \( ca_1...a_k, \; k \geq 0 \) without back-tracking (\( aba \)); ii) edges : \( (ca_1...a_{k+1}, ca_1...a_k) \).

*All these graphs are the same up to isomorphism.*
For $c$ vertex, the **universal cover** $\mathcal{U}_c(G)$ of $G$ is the graph s.t.: i) vertices: $ca_1...a_k$, $k \geq 0$ without back-tracking ($aba$); ii) edges: $(ca_1...a_{k+1}, ca_1...a_k)$.

*All these graphs are the same up to isomorphism.*

**Ex:**

![Diagram of $G$ and $\mathcal{U}_c(G)$]
When $G$ is square free, every pair $(c, z)$, $z \in \mathbb{Z}^2$ defines a 'natural' function from $X_G$ to $X_{U_c(G)}$:

$$y \in X_{U_c(G)}$$

$$x \in X_G$$
When $G$ is square free, every pair $(c, z), z \in \mathbb{Z}^2$ defines a ’natural’ function from $X_G$ to $X_{\mathcal{U}_c(G)}$:

$$y \in X_{\mathcal{U}_c(G)}$$

$$x \in X_G$$

where $p_a$ is a path of smallest length from $c$ to $a$. 
When $G$ is square free, every pair $(c, z)$, $z \in \mathbb{Z}^2$ defines a ’natural’ function from $X_G$ to $X_{U_c(G)}$:

$$y \in X_{U_c(G)} \quad \text{and} \quad x \in X_G$$

where $p_a$ is a path of smallest length from $c$ to $a$. 
When $G$ is square free, every pair $(c, z)$, $z \in \mathbb{Z}^2$ defines a ‘natural’ function from $X_G$ to $X_{U_c(G)}$:

\[ y \in X_{U_c(G)} \quad \text{and} \quad x \in X_G \]

where $p_a$ is a path of smallest length from $c$ to $a$. 
When $G$ is square free, every pair $(c, z)$, $z \in \mathbb{Z}^2$ defines a ‘natural’ function from $X_G$ to $X_{\mathcal{U}_c(G)}$:

$$y \in X_{\mathcal{U}_c(G)} \quad \text{and} \quad x \in X_G$$

where $p_a$ is a path of smallest length from $c$ to $a$. 
When \( G \) is square free, every pair \((c, z), z \in \mathbb{Z}^2\) defines a 'natural' function from \( X_G \) to \( X_{U_c(G)} \):

\[
y \in X_{U_c(G)} \quad \quad \quad \quad x \in X_G
\]

where \( p_a \) is a path of smallest length from \( c \) to \( a \).
When $G$ is square free, every pair $(c, z), z \in \mathbb{Z}^2$ defines a ‘natural’ function from $X_G$ to $X_{\mathcal{U}_c(G)}$:

$$y \in X_{\mathcal{U}_c(G)} \quad \text{and} \quad x \in X_G$$

where $p_a$ is a path of smallest length from $c$ to $a$. 
When $G$ is square free, every pair $(c, z)$, $z \in \mathbb{Z}^2$ defines a 'natural' function from $X_G$ to $X_{U_c(G)}$:

$$y \in X_{U_c(G)} \quad x \in X_G$$

where $p_a$ is a path of smallest length from $c$ to $a$. 
Lemma: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.
**Lemma**: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.

**Proof**: every infinite row can appear below to its right shift.
Lemma: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.

Proof: every infinite row can appear below to its right shift.
Lemma: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.

Proof: every infinite row can appear below to its right shift.
Lemma: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.

Proof: every infinite row can appear below to its right shift.
Lemma: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.

Proof: every infinite row can appear below to its right shift.
Lemma: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.

Proof: every infinite row can appear below to its right shift.
**Lemma**: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.

**Proof**: every infinite row can appear below to its right shift.
Lemma: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.

Proof: every infinite row can appear below to its right shift.
Lemma: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.

Proof: every infinite row can appear below to its right shift.
Lemma: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.

Proof: every infinite row can appear below to its right shift.
**Lemma**: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.

**Proof**: every infinite row can appear below to its right shift.
**Lemma**: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.

*Proof*: every infinite row can appear below to its right shift.
Lemma: a Hom shift is transitive if and only if $G$ is connected. For $G$ connected it is at most $O(n)$-transitive.

Proof: every infinite row can appear below to its right shift.
**Theorem** [B. Marcus, N. Chandgotia] : when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.
Theorem [B. Marcus, N. Chandgotia]: When $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

Proof: 1. The universal cover is a finite graph. This implies that $G$ is a finite tree.

```
abac|d|ce
```

```
\begin{tikzpicture}
  \node (a) at (0,0) {a};
  \node (b) at (1,1) {b};
  \node (c) at (0,-1) {c};
  \node (d) at (0.5,-1.5) {d};
  \node (e) at (1,-2) {e};

  \draw (a) -- (b);
  \draw (a) -- (c);
  \draw (c) -- (d);
  \draw (d) -- (e);
\end{tikzpicture}
```
**Theorem** [B. Marcus, N. Chandgotia]: when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

**Proof**: 1. The universal cover is a finite graph. This implies that $G$ is a finite tree.
**Theorem** [B. Marcus, N. Chandgotia]: when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

**Proof:** 1. The universal cover is a finite graph. This implies that $G$ is a finite tree.
**Theorem** [B. Marcus, N. Chandgotia]: when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

**Proof**: 1. The universal cover is a finite graph. This implies that $G$ is a finite tree.
**Theorem** [B. Marcus, N. Chandgotia]: when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

**Proof:** 1. The universal cover is a finite graph. This implies that $G$ is a finite tree.
**Theorem** [B. Marcus, N. Chandgotia]: when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

**Proof**: 1. The universal cover is a finite graph. This implies that $G$ is a finite tree.
**Theorem** [B. Marcus, N. Chandgotia]: when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

**Proof**: 1. The universal cover is a finite graph. This implies that $G$ is a finite tree.
**Theorem**[B.Marcus, N.Chandgotia] : when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

**Proof** : 1. The universal cover is a finite graph. This implies that $G$ is a finite tree.
**Theorem** [B. Marcus, N. Chandgotia]: when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

*Proof*: 2. The universal cover is an infinite graph.
Theorem [B. Marcus, N. Chandgotia]: when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

Proof: 2. The universal cover is an infinite graph.

For $n \geq 0$, consider some non-backtracking path $u = a_1 \ldots a_{2n+1}$, and $v = (a_1 a_2)^n a_1$. 
**Theorem** [B. Marcus, N. Chandgotia]: when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

**Proof:** 2. The universal cover is an infinite graph.

For $n \geq 0$, consider some non-backtracking path $u = a_1 \ldots a_{2n+1}$, and $v = (a_1 a_2)^n a_1$.

Assume $u, v$ can be glued at distance $< n$. 
**Theorem**[B. Marcus, N. Chandgotia] : when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

**Proof** : 2. The universal cover is an infinite graph.

For $n \geq 0$, consider some non-backtracking path $u = a_1...a_{2n+1}$, and $v = (a_1a_2)^na_1$.

Assume $u, v$ can be glued at distance $< n$.

$$x \in X_G$$
**Theorem** [B. Marcus, N. Chandgotia]: when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

**Proof:** 2. The universal cover is an infinite graph.

For $n \geq 0$, consider some non-backtracking path $u = a_1...a_{2n+1}$, and $v = (a_1a_2)^na_1$.

Assume $u, v$ can be glued at distance $< n$.

\[ x \in X_G \]
**Theorem**[B. Marcus, N. Chandgotia]: when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

**Proof**: 2. The universal cover is an infinite graph.

For $n \geq 0$, consider some non-backtracking path $u = a_1 \ldots a_{2n+1}$, and $v = (a_1 a_2)^n a_1$.

Assume $u, v$ can be glued at distance $< n$.

\[
\begin{align*}
&x \in X_G \\
&y \in U_{a_1}(G)
\end{align*}
\]
**Theorem** [B. Marcus, N. Chandgotia]: when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

**Proof:** 2. The universal cover is an infinite graph.

For $n \geq 0$, consider some non-backtracking path $u = a_1 \ldots a_{2n+1}$, and $v = (a_1 a_2)^n a_1$.

Assume $u, v$ can be glued at distance $< n$.

\[ x \in X_G \quad y \in \mathcal{U}_{a_1}(G) \]
Theorem[B.Marcus, N.Chandgotia]: when $G$ is square-free, $X_G$ is $\Theta(1)$-transitive or $\Theta(n)$-transitive.

Proof: 2. The universal cover is an infinite graph.

For $n \geq 0$, consider some non-backtracking path $u = a_1\ldots a_{2n+1}$, and $v = (a_1 a_2)^n a_1$.

Assume $u, v$ can be glued at distance $< n$.

The paths $p$ and $q$ have to be equal in the universal cover, which is impossible.
Our results
Pavlov and Schraudner’s conjecture

**Conjecture**[R. Pavlov, M. Schraudner]: $\Theta(1)$ and $\Theta(n)$ are the only transitivity classes for Hom shifts.
Pavlov and Schraudner’s conjecture

**Conjecture** [R. Pavlov, M. Schraudner]: $\Theta(1)$ and $\Theta(n)$ are the only transitivity classes for Hom shifts.

**Counterexample** [S. Gangloff, B. Hellouin, P. Oprocha]: The following graph $K$ provides a counter-example:

Indeed, we proved that $X_K$ is $\Theta(\log(n))$-transitive.
Proof: 1. $X_K$ is at least $\log(n)$-transitive.
Proof: 1. $X_K$ is at least $\log(n)$-transitive.
Proof: 1. \( X_K \) is at least \( \log(n) \)-transitive.
Proof: 1. $X_K$ is at least $\log(n)$-transitive.
Proof: 1. $X_K$ is at least $\log(n)$-transitive.
Proof: 1. $X_K$ is at least $\log(n)$-transitive.
Proof: 1. $X_K$ is at least $\log(n)$-transitive.

The shift is forced on the remainder of $w$. 
Proof: 1. $X_K$ is at least $\log(n)$-transitive.

The shift is forced on the remainder of $w$. 

$$\text{without } c$$
Proof: 1. $X_K$ is at least $\log(n)$-transitive.

The shift is forced on the remainder of $w$. 

$\text{without } c$
Proof: 1. $X_K$ is at least $\log(n)$-transitive.

The shift is forced on the remainder of $w$. 

\[ c^n \] without $c$
Proof: 1. $X_K$ is at least $\log(n)$-transitive.

The shift is forced on the remainder of $w$. 

\[ \text{without } c \]
Proof: 1. $X_K$ is at least $\log(n)$-transitive.

The shift is forced on the remainder of $w$. 

\[ \text{without } c \]
Proof: 1. $X_K$ is at least $\log(n)$-transitive.

The shift is forced on the remainder of $w$.

For $\mu_c(w)$ maximal size of a $c$-block in $w$: $\mu_c(w) \geq \frac{1}{2} \mu_c(c^n) - 3$. 
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

i) Procedure to smash down a simple cycle in $K$:
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

i) Procedure to smash down a simple cycle in $K$:
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

i) Procedure to smash down a simple cycle in $K$: 

![Diagram of a graph with arrows indicating the procedure to smash down a simple cycle in $K$.]
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

i) Procedure to smash down a simple cycle in $K$:
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

i) Procedure to smash down a simple cycle in $K$:
Proof: 2. $\chi_K$ is at most $\log(n)$-transitive.

i) Procedure to smash down a simple cycle in $K$:
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

i) Procedure to smash down a simple cycle in $K$:
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

i) Procedure to smash down a simple cycle in $K$:
Proof: 2. $\chi_K$ is at most $\log(n)$-transitive.

i) Procedure to smash down a simple cycle in $K$:
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

i) Procedure to smash down a simple cycle in $K$:
Proof: 2. $\chi_K$ is at most $\log(n)$-transitive.

i) Procedure to smash down a simple cycle in $K$:

Expansion of backtracking parts:
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

i) Procedure to smash down a simple cycle in $K$:

![Diagram](image-url)

Expansion of backtracking parts:
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

ii) How to smash down an iterate of a cycle:

\[
\sigma \quad c \quad c \quad c \quad \cdots \quad c \quad \cdots \quad c \quad c \quad c
\]
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

ii) How to smash down an iterate of a cycle:

\[
\begin{array}{cccccccc}
  c & c & c & \cdots & c & \cdots & c & c \\
\end{array}
\]

\[
\begin{array}{cccccccc}
  c & c & c & \cdots & * & \cdots & c & c \\
\end{array}
\]
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

ii) How to smash down an iterate of a cycle:

\[
\begin{array}{cccccccc}
  c & c & c & \cdots & c & \cdots & c & c \\
  & & & & & & & \\
  c & c & c & \cdots & \ast & \cdots & c & c \\
  c & c & c & \cdots & \ast & \cdots & c & c \\
\end{array}
\]
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

ii) How to smash down an iterate of a cycle:

\[
\begin{array}{cccccccc}
  c & c & c & \cdots & c & c & c \\
  \downarrow & & & & & & \\
  c & c & c & \cdots & * & \cdots & c & c \ \\
  c & c & c & \cdots & * & \cdots & c & c \\
\end{array}
\]
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

ii) How to smash down an iterate of a cycle:

\[
\begin{array}{c}
\text{c c c \cdots c \cdots c c c} \\
\downarrow \\
\text{c c c \cdots \ast \cdots c c c} \\
\text{c c c \cdots \ast \cdots c c c}
\end{array}
\]
Proof: 2. \( X_K \) is at most \( \log(n) \)-transitive.

ii) How to smash down an iterate of a cycle:

\[
\begin{array}{ccccccccccc}
C & C & C & \cdots & C & \cdots & C & C & C \\
\downarrow & & & & & & & & & & \\
C & C & C & \cdots & * & \cdots & C & C & C \\
C & C & C & \cdots & * & \cdots & C & C & C \\
\end{array}
\]

\[
C \xrightarrow{\sigma} \ast \xleftarrow{\sigma} \cdots C \xrightarrow{\sigma} \cdots C \xleftarrow{\sigma} C
\]
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

ii) How to smash down an iterate of a cycle:

$$
\begin{array}{cccccc}
\text{c} & \text{c} & \text{c} & \cdots & \text{c} & \cdots & \text{c} & \text{c} & \text{c} \\
\downarrow \\
\begin{array}{cccccc}
\text{c} & \text{c} & \text{c} & \cdots & \ast & \cdots & \text{c} & \text{c} & \text{c} \\
\text{c} & \text{c} & \text{c} & \cdots & \ast & \cdots & \text{c} & \text{c} & \text{c} \\
\end{array} \\
\begin{array}{cccccc}
\text{c} & \text{c} & \text{c} & \cdots & \ast & \cdots & \text{c} & \text{c} & \text{c} \\
\sigma & \cdots & \ast & \cdots & \sigma & \\
\end{array}
\end{array}
$$
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

ii) How to smash down an iterate of a cycle:

$$
\begin{array}{c}
\sigma \quad \sigma \\
\downarrow \\
\sigma \\
\end{array}
$$
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

ii) How to smash down an iterate of a cycle:

\[
\begin{array}{ccccccc}
\; & c & c & c & \cdots & c & c & c \\
\downarrow & & & & & & & \\
\; & c & c & c & \cdots & * & \cdots & c & c & c \\
\; & c & c & c & \cdots & * & \cdots & c & c & c \\
\; & c & c & c & \cdots & * & \cdots & c & c & c \\
\end{array}
\]
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

ii) How to smash down an iterate of a cycle:

\[
\begin{array}{cccccccccc}
  c & c & c & \cdots & c & \cdots & c & c & c \\
  & & & & & & & & \\
  \downarrow & & & & & & & & \\
  c & c & c & \cdots & \ast & \cdots & c & c & c \\
  c & c & c & \cdots & \ast & \cdots & c & c & c \\
  & & & & & & & & \\
  c & c & c & \cdots & \ast & \cdots & c & c & c \\
  & & & & & & & & \\
  \sigma & & & \cdots & & \sigma & & & \\
  \downarrow & & & & & & & & \\
  c & c & c & \cdots & t & \cdots & c & c & c \\
\end{array}
\]
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

ii) How to smash down an iterate of a cycle:

\[
c \quad c \quad c \quad \cdots \c c \quad \cdots \c c \quad c \\
\downarrow \\
\boxed{c \quad c \quad c \quad \cdots \ast \quad \cdots \c c \quad c \quad c}
\]

\[
c \quad c \quad c \quad \cdots \ast \quad \cdots \c c \quad c \\[\sigma\]
\downarrow \\
\c c \quad c \quad \cdots \ t \quad \cdots \c c \quad c \\
\c \ast \quad c \quad \cdots \ t' \quad \cdots \c \ast \quad c \\[\sigma\]
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

iii) How to smash down any cycle:

![Diagram of a cycle](image)
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

iii) How to smash down any cycle:

\[\text{Diagram of cycles to illustrate the process.}\]
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

iii) How to smash down any cycle:

![Diagram of cycles and their transformations](attachment:image.png)
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

iii) How to smash down any cycle:

![Diagrams of cycles and operations to smash them down]
Proof: 2. $X_K$ is at most $\log(n)$-transitive.

iii) How to smash down any cycle:

```
  _______  _______  _______
 /       |         |       \
|   X    |   X      |   X   |
|_______  |_______   |_______|
```

iv) Every path of even length can be transformed into a cycle in a bounded number of steps.
Quaternary cover:

Square equivalence for non-backtracking paths:
Quaternary cover:

Square equivalence for non-backtracking paths:

Quaternary cover: quotient of the universal cover by square equivalence.
Some examples of quaternary cover

\[
\begin{array}{c}
\quad \\
\begin{array}{ccc}
\quad & \quad & \\
\end{array}
\end{array}
\]
Square dismantlability

**Decomposability** : a cycle is decomposable whenever it is square equivalent to a trivial cycle.
Square dismantlability

**Decomposability** : a cycle is decomposable whenever it is square equivalent to a trivial cycle.

**Dismantlability** : a graph $G$ is square-dismantlable whenever every simple cycle is decomposable.
Square dismantlability

Decomposability: a cycle is decomposable whenever it is square equivalent to a trivial cycle.

Dismantlability: a graph $G$ is square-dismantlable whenever every simple cycle is decomposable.

Lemma: the quaternary cover of a graph is always square-dismantlable.
Generalization

**Theorem** [S. Gangloff, B. Hellouin, P. Oprocha] : Whenever the graph $G$ is *square dismantlable*, $X_G$ is $O(\log(n))$-transitive.
Generalization

**Theorem** [S. Gangloff, B. Hellouin, P. Oprocha] : Whenever the graph $G$ is *square dismantlable*, $X_G$ is $O(\log(n))$-transitive.

As a consequence :

**Theorem** [S. Gangloff, B. Hellouin, P. Oprocha] : Whenever the graph $G$ has a finite quaternary cover, $X_G$ is $O(\log(n))$-transitive.
**Generalization**

**Theorem**[S. Gangloff, B. Hellouin, P. Oprocha] : Whenever the graph $G$ is *square dismantlable*, $X_G$ is $O(\log(n))$-transitive.

As a consequence :

**Theorem**[S. Gangloff, B. Hellouin, P. Oprocha] : Whenever the graph $G$ has a finite quaternary cover, $X_G$ is $O(\log(n))$-transitive. Furthermore :

**Theorem**[S. Gangloff, B. Hellouin, P. Oprocha] : Whenever the quaternary cover of $G$ is infinite, $X_G$ is $\Theta(n)$-transitive.
Further research
**Middle term goal**: Prove a similar result for the class of bidimensional SFT, or tools to produce examples between \( \Theta(\log(n)) \) and \( \Theta(n) \).

**Long term goal**: What happens to the computability of entropy between \( \Theta(\log(n)) \) and \( \Theta(n) \) for bidimensional SFT?

Some natural short-term questions:

1. Is there an algorithm which decides, provided \( G \), if its quaternary cover is finite or infinite?
2. What happens when \( G \) is oriented?
3. For shifts of finite type corresponding to graphs \( G_1, G_2 \) isomorphic?