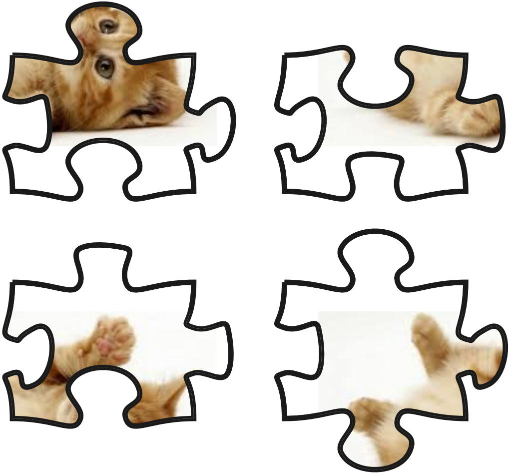
Higman type theorems for subshifts

Emmanuel Jeandel, Pascal Vanier

Laboratoire d'Algorithmique Complexité et Logique, UPEC

Puzzle

What could this be ?



Given an infinite number of puzzle pieces can we tile infinitely in all directions with them ?

Answer

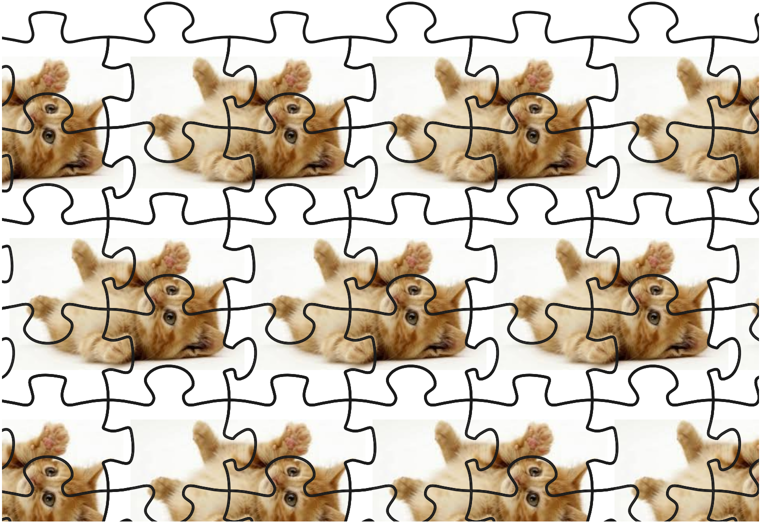
This was a very hard puzzle...

Answer

This was a very hard puzzle...

Answer

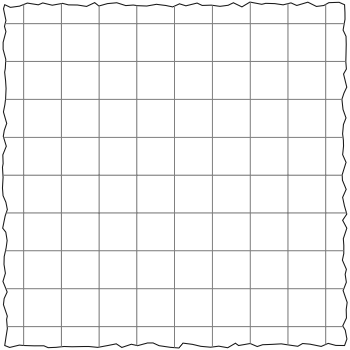
This was a very hard puzzle...



Subshifts and subshifts of finite type

A **finite** alphabet:

$$\Sigma = \{\blacksquare, \blacksquare\}$$

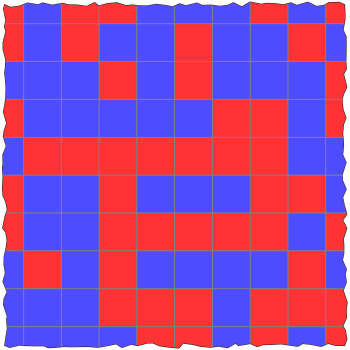


Subshifts and subshifts of finite type

A **finite** alphabet:

$\Sigma = \{$,$\}$

A **tiling** or **configuration** is a coloring of $\mathbb{Z}^d$:



Subshifts and subshifts of finite type

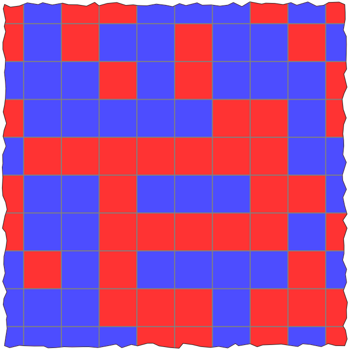
A finite alphabet:

$$\Sigma = \{$$  $$\}$$

A finite number of forbidden patterns:

$$\mathcal{F} = \Big\{$$  $$\Big\}$$

A **tiling** or **configuration** is a coloring of $\mathbb{Z}^d$:



Subshifts and subshifts of finite type

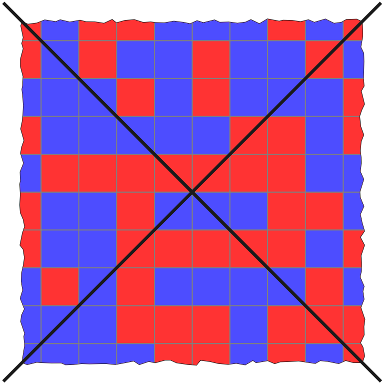
A **finite** alphabet:

$\Sigma = \{$  $\}$

A **finite** number of forbidden patterns:

$\mathcal{F} = \Bigg\{$  $\Bigg\}$

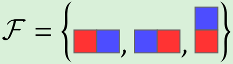
A **tiling** or **configuration** is a coloring of $\mathbb{Z}^d$:



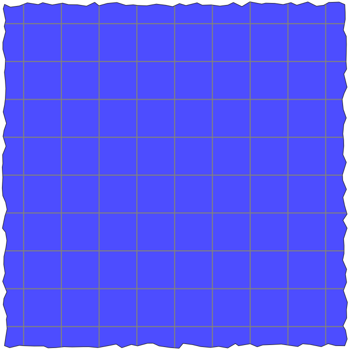
Subshifts and subshifts of finite type

A **finite** alphabet:

A **finite** number of forbidden patterns:



A **tiling** or **configuration** is a coloring of $\mathbb{Z}^d$:



Subshifts and subshifts of finite type

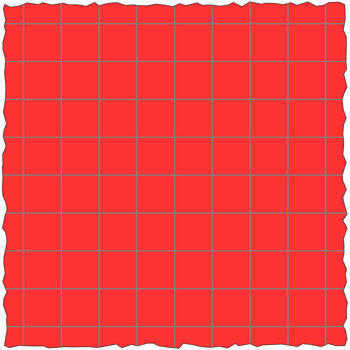
A **finite** alphabet:

$$\Sigma = \{\ \}$$

A **finite** number of forbidden patterns:

$$\mathcal{F} = \{\ \}$$

A **tiling** or **configuration** is a coloring of $\mathbb{Z}^d$:



Subshifts and subshifts of finite type

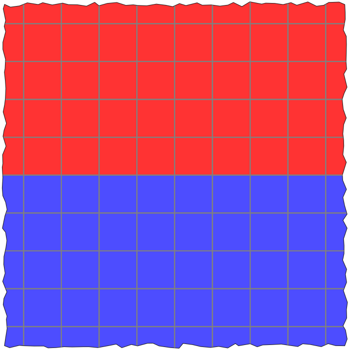
A **finite** alphabet:

$\Sigma = \{$  $\}$

A **finite** number of forbidden patterns:

$\mathcal{F} = \Big\{$  $\Big\}$

A **tiling** or **configuration** is a coloring of $\mathbb{Z}^d$:



Subshifts and subshifts of finite type

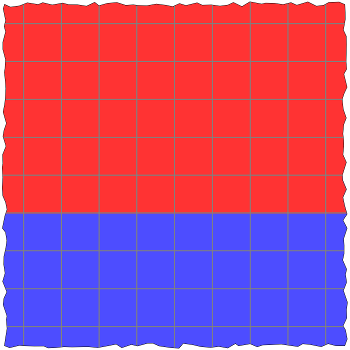
A **finite** alphabet:

$$\Sigma = \{$$  $$\}$$

A **finite** number of forbidden patterns:

$$\mathcal{F} = \Big\{$$  $$\Big\}$$

A **tiling** or **configuration** is a coloring of $\mathbb{Z}^d$:



Subshifts and subshifts of finite type

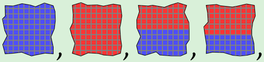
A **finite** alphabet:

$$\Sigma = \{$$  $$\}$$

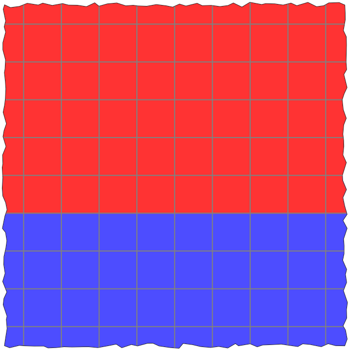
A **finite** number of forbidden patterns:

$$\mathcal{F} = \Bigg\{$$  $$\Bigg\}$$

Subshift of finite type (SFT): set of configurations avoiding $\mathcal{F}$. We note $\mathcal{X}_{\mathcal{F}}$:

$$\mathcal{X}_{\mathcal{F}} = \Bigg\{$$  $$, \ldots \Bigg\}$$

A **tiling** or **configuration** is a coloring of $\mathbb{Z}^d$:



Subshifts and subshifts of finite type

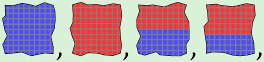
A **finite** alphabet:

$$\Sigma = \{$$  $$\}$$

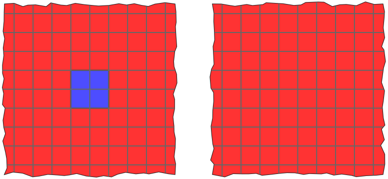
A **finite** number of forbidden patterns:

$$\mathcal{F} = \Bigg\{$$  $$\Bigg\}$$

Subshift of finite type (SFT): set of configurations avoiding $\mathcal{F}$. We note $\mathcal{X}_{\mathcal{F}}$:

$$\mathcal{X}_{\mathcal{F}} = \Bigg\{$$  $$, \dots \Bigg\}$$

The **family** may also be **infinite** we then talk about **subshifts**.



Another example

Let $\mathcal{F} = \{ab, ba\}$ and $\Sigma = \{a, b\}$:

$X_{\mathcal{F}} =$

$$\cdots aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa \cdots$$

$$\cdots bbbbbbbbbbbbbbbbbbbbbbbbbbbb \cdots$$

Finitely generated groups

Finitely generated group: S finite set of generators and R a set of relations, $\langle S \mid R \rangle$ is the largest group generated by S in which all of R holds.

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

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

Examples of groups and subshifts

Example 1: groups

- $G = \langle a, b \mid \rangle$: the free group with two generators.
- $G = \langle a, b \mid ab, ba \rangle$ Reduced words of the form: a^n, b^n

Example 2: subshifts

- $\Sigma = \{a, b\}$ and $\mathcal{F} = \emptyset$: the full shift over two symbols.
- $\Sigma = \{a, b\}$ and $\mathcal{F} = \{ab, ba\}$ Configurations: ${}^{\omega}b^{\omega}, {}^{\omega}a^{\omega}$

Notations

Alphabet and

- *Set of relations* defines a group.
- *Set of forbidden patterns* defines a subshift.

Group:

$$\langle S \mid R \rangle$$

Subshift of dimension d:

$$\langle \Sigma \mid \mathcal{F} \rangle^d$$

Language/Word problem

More generally:

Word Problem:
$WP(G) = \{w \mid w = 1_G\}$

Complement of the language:
$\mathcal{L}(X)^c = \{m \mid \forall x \in X, m \not\sqsubseteq x\}$

$WP(G)$ is recursively enumerable from the set of relations.

$\mathcal{L}(X)^c$ is recursively enumerable from the set of forbidden patterns.

$$G = \langle S_G \mid WP(G) \rangle$$

$$X = \left\langle \Sigma_X \,\middle|\, \mathcal{L}(X)^c \right\rangle$$

Similar definitions

Finitely many relations/forbidden patterns

- Subshifts of finite type
- Finitely presented groups

Recursively enumerable set of relations/forbidden patterns

- Effective subshifts
- Recursively presented groups

$WP(G)$ and $\mathcal{L}(X)^c$ are recursively enumerable in both cases.

1. Analogies
2. Higman embedding theorem
3. Relative Higman embedding theorem
4. Boone-Higman-Thompson theorem
5. Conclusion

Adding relations

Let $X = \langle A \mid R \rangle$ and $Y = \langle A \mid R \cup Q \rangle$:

Groups:
Y is a quotient subgroup of X by some normal subgroup.

Subshifts:
Y is a subshift of X.

Adding relations

$T = \langle A \mid R \rangle$ becomes trivial if we add any relation/pattern to R.

Groups:
T is simple.

Subshifts:
T is minimal.

Restricting

Let $T = \langle A \mid R \rangle$ and $S = \langle B \mid R \rangle$ with $B \subsetneq A$.

Groups:
S is a subgroup of T

Not all subgroups are of this form.

Subshifts:
$\mathcal{L}(S) = \mathcal{L}(T) \cap B^{*d}$

Restricting: example

$$X = \left\langle a, b \;\middle|\; \begin{pmatrix} a & a \end{pmatrix}, \begin{pmatrix} b & b \end{pmatrix}, \begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} b \\ a \end{pmatrix} \right\rangle^2$$

$$\begin{matrix} \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & a & b & a & b & a & b & \dots \\ \dots & a & b & a & b & a & b & \dots \\ \dots & a & b & a & b & a & b & \dots \\ \dots & a & b & a & b & a & b & \dots \\ \dots & a & b & a & b & a & b & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{matrix}$$

S s.t. $\mathcal{L}(S) = \mathcal{L}(X) \cap \{a, b\}^{*1}$:

$$\cdots abababababababababababab \cdots$$

S' s.t. $\mathcal{L}(S') = \mathcal{L}(X) \cap \{a\}^{*1}$:

$$\emptyset$$

Free product

Definition The free product of $F = \langle A \mid R \rangle$ and $G = \langle B \mid Q \rangle$ is:

$$F * G = \langle A \cup B \mid R \cup Q \rangle$$

Remark Adding symbols corresponds to the free product by a free group/full shift.

Free product: example

$$X = \left\langle a, b \;\middle|\; \begin{pmatrix} a & a \end{pmatrix}, \begin{pmatrix} b & b \end{pmatrix}, \begin{pmatrix} a \\ a \end{pmatrix}, \begin{pmatrix} b \\ b \end{pmatrix} \right\rangle^2$$

$$Y = \left\langle c, d \;\middle|\; \begin{pmatrix} c & d \end{pmatrix}, \begin{pmatrix} d, c \end{pmatrix}, \begin{pmatrix} c \\ d \end{pmatrix} \right\rangle^2$$

$$\begin{array}{cccccccccccccc}
d & d & d & d & d & d & d & d & d & d & d & d & d & d \\
d & d & d & d & d & d & d & d & d & d & d & d & d & d \\
c & c & c & c & c & c & c & c & c & c & c & c & a & d \\
b & c & b & c & b & c & c & b & a & b & a & c & b & d \\
a & b & a & b & a & c & c & a & b & a & b & c & a & d \\
b & a & b & a & b & c & c & b & a & b & a & c & b & d \\
a & b & a & b & a & c & c & c & c & c & c & c & a & d \\
b & a & b & a & b & c & c & c & c & c & c & c & b & d \\
a & b & a & b & a & c & c & c & c & c & c & c & a & d
\end{array}$$

$$\in X * Y$$

The rosetta stone

Group G	**Subshift** X
Group with n generators	Subshift on n symbols
Free group with n generators	Full shift on n symbols
Word problem $WP(G)$	co-language $\mathcal{L}(X)^c$
Finitely presented group	SFT
Recursively presented group	Effectively closed subshift
Simple group	Minimal subshift
G_1 is a quotient of G_2	$X_1 \subseteq X_2$
$G_1 \subseteq G_2$ by restricting the generators	$\mathcal{L}(X_1) = \mathcal{L}(X_2) \cap \Sigma^{*d}$

Some decidability theorems

Theorem It is **undecidable** whether a **f.p. group is trivial**.

Theorem **f.p. simple groups** have **computable word problem**.

Theorem It is **undecidable** whether an **SFT is empty**.

Theorem A **minimal SFT** has **computable language**

Restrictions: powerfullness

Take X an effective subshift on some alphabet Σ.

There exists a **computable family of forbidden patterns** $\mathcal{F}$ generating it.

$X' = \langle \Sigma \cup \{\#\} \mid \mathcal{F} \rangle$ has a computable language:

Any pattern on Σ containing no pattern of $\mathcal{F}$ may appear between #.

The restriction of X' to Σ is X which may not have a computable language.

1. Analogies

2. **Higman embedding theorem**

3. Relative Higman embedding theorem

4. Boone-Higman-Thompson theorem

5. Conclusion

Higman embedding theorem for groups

Theorem [Higman 1961] G is **recursively presented** iff it is a **subgroup** of some **finitely presented group** H.

The proof is stronger actually: G is obtained by restriction of H.

Statement reminiscent of :

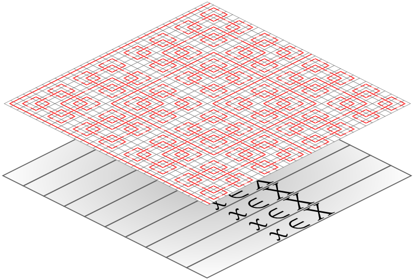
Theorem [Hochman 2009,Aubrun Sablik 2013,Durand Romashchenko Shen 2011]
A **subshift** of dimension d is **effective** iff it is a **projective subaction** of some **SFT** of dimension $d+1$.

Higman embedding theorem for subshifts

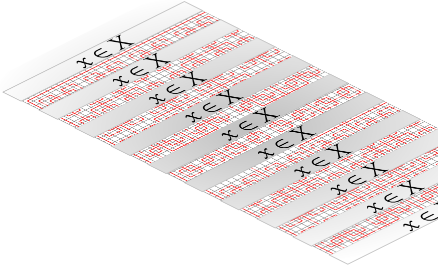
Theorem X is an effective subshift iff it can be obtained by restriction of some SFT Y.

Proof.

AS 2013, DRS 2011 proof:



Small modification:



X is then obtaining by restricting Y to X's alphabet. $\square$

1. Analogies
2. Higman embedding theorem
3. **Relative Higman embedding theorem**
4. Boone-Higman-Thompson theorem
5. Conclusion

Relative embedding theorem for groups

Definition G is finitely presented over H if it can be obtained by adding finitely many generators and relations to H.

Definition $A \leq_e B$ iff there exists $f : \mathbb{N} \times \mathbb{N} \to \mathbb{P}_{finite}(\mathbb{N})$ computable s.t.:

$$x \in A \Leftrightarrow \exists n \in \mathbb{N}, f(n,x) \subseteq B$$

A is uniformly enumerable from any enumeration of B.

Theorem [Higman Scott 1988] K is a subgroup of a finitely presented group over G iff $WP(K) \leq_e WP(G)$.

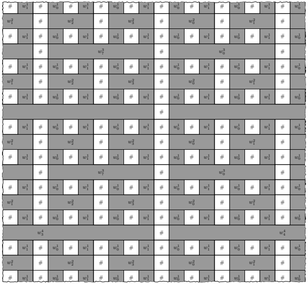
Relative embedding theorem for subshifts

Definition X is an SFT over Y if it can be obtained by adding finitely many symbols and forbdidden patterns to Y.

Theorem X is a restriction of an SFT over Y iff $\mathcal{L}(X)^c \leq_e \mathcal{L}(Y)^c$.

Proof idea

Make a subshift containing $\mathcal{L}(Y)$ in at least one configuration:



Insert lines of X inbetween the lines: if a pattern m is not allowed, then $\exists n, f(n,m) \notin \mathcal{L}(Y)$, forbid that.

Boone-Higman-Thompson theorem

For groups:

Theorem [Boone-Higman 1974, Thompson 1980]
$WP(G)$ **is computable** iff G is a **subgroup of a simple finitely presented group**.

For subshifts:

Theorem $\mathcal{L}(X)$ **is computable** iff X is a **restriction of a minimal effective subshift**.

Proof

Clear:

$$X \text{ minimal effectively closed} \Rightarrow \mathcal{L}(X) \cap \Sigma^* \text{ computable.}$$

Not so clear:

$$\mathcal{L}(X) \text{ computable} \Rightarrow \text{there exists } X' \text{ minimal effectively closed such that } \mathcal{L}(X) = \mathcal{L}(X') \cap \Sigma^*.$$

Proof: Two steps

Theorem A subshift X **has a computable language** iff it is the **restriction of some minimal effective subshift** Y.

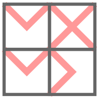
Theorem [Durand Romashchenko 2018] An **minimal effective subshift** can be realized as a **subaction of some minimal SFT**.

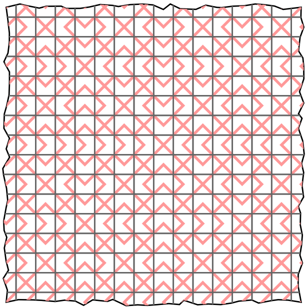
Minimality

Definition A subshift X is **minimal** iff there is **no subshift Y s.t. $Y \subsetneq X$**.

For each pattern there is a *window* in which it always appears.

Example :



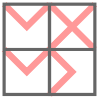


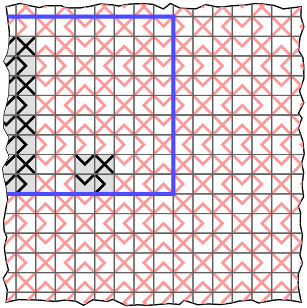
Minimality

Definition A subshift X is **minimal** iff there is **no subshift Y s.t. $Y \subsetneq X$**.

For each pattern there is a *window* in which it always appears.

Example :



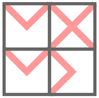


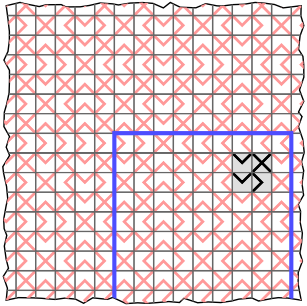
Minimality

Definition A subshift X is **minimal** iff there is **no subshift Y s.t. $Y \subsetneq X$**.

For each pattern there is a *window* in which it always appears.

Example :



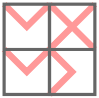


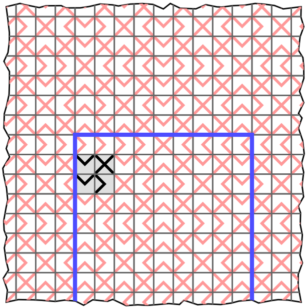
Minimality

Definition A subshift X is **minimal** iff there is **no subshift Y s.t. $Y \subsetneq X$**.

For each pattern there is a *window* in which it always appears.

Example :



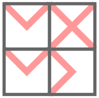


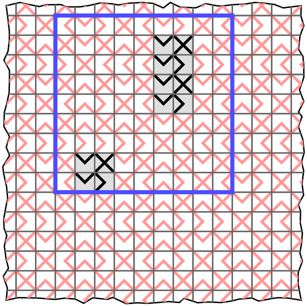
Minimality

Definition A subshift X is **minimal** iff there is **no subshift Y s.t. $Y \subsetneq X$**.

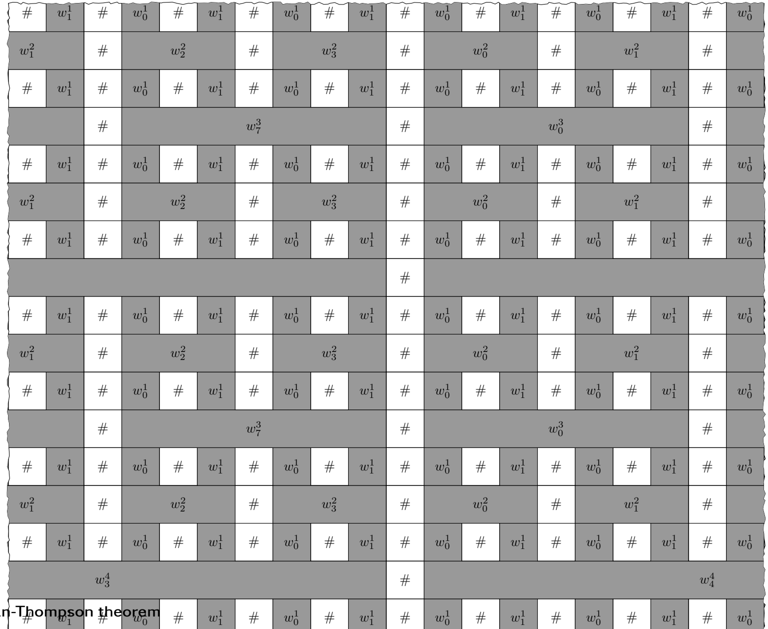
For each pattern there is a *window* in which it always appears.

Example :

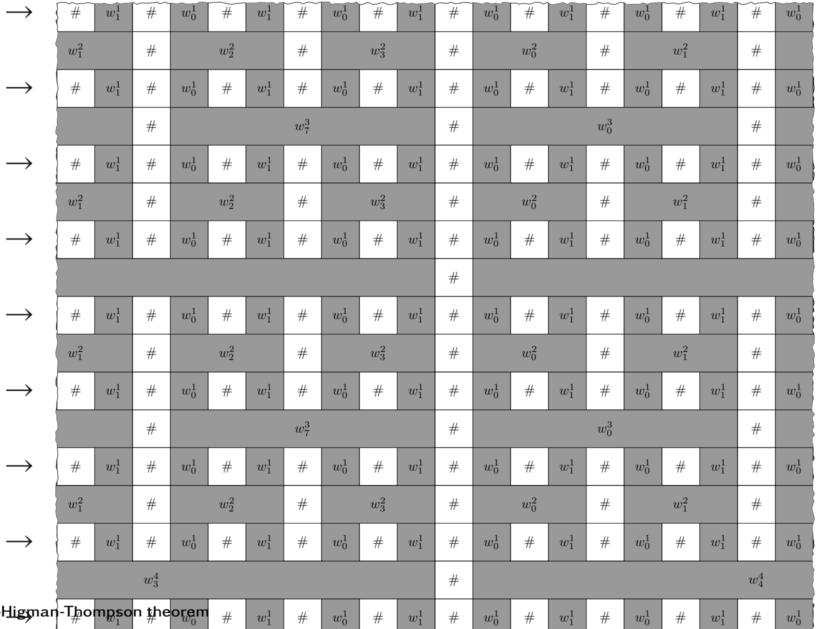




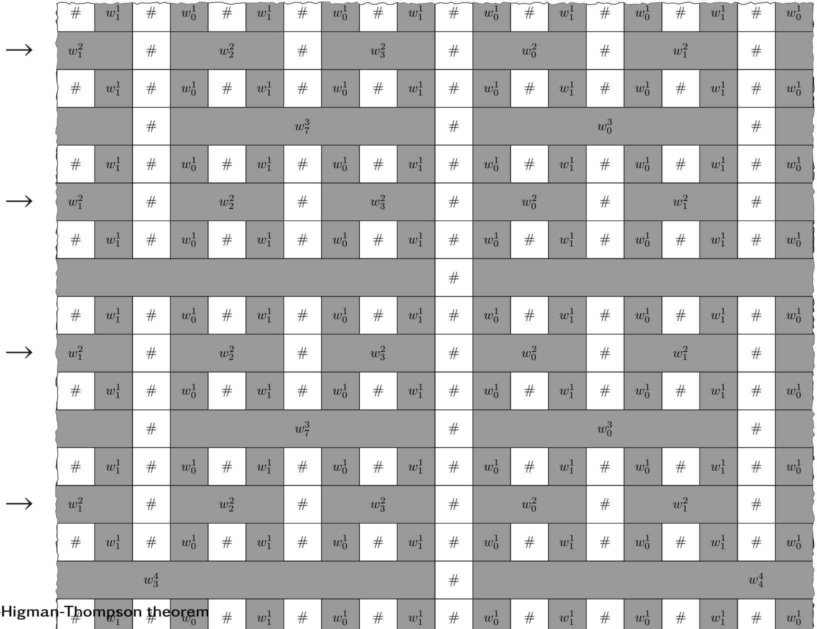
Proof: minimal effective construction



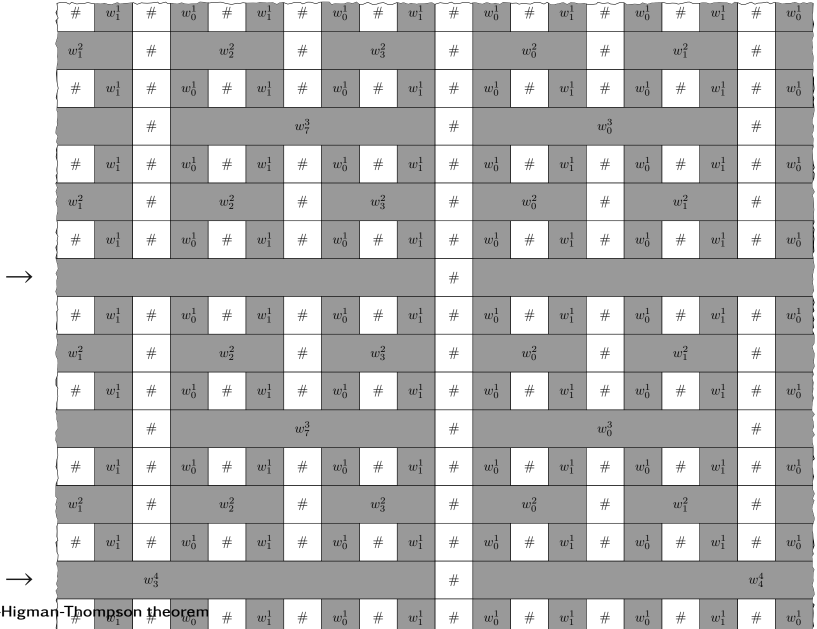
Proof: minimal effective construction



Proof: minimal effective construction



Proof: minimal effective construction



Proof: minimal effective construction

Let's make a computable configuration:

- Every row has a level l and contains all pairs of words of $\mathcal{L}(X)$ of length p_l separated by # periodically.
- A row of level l appears every 2^l lines and contains words of length p_l the period of the previous level.
- The row of level 0 appears at most once and contains no #.

The subshift thus generated is minimal and effective.

1. Analogies
2. Higman embedding theorem
3. Relative Higman embedding theorem
4. Boone-Higman-Thompson theorem
5. Conclusion

Conclusion

- Dictionnary between subshifts/groups: not perfect, room for improvement.
- Intuition for proving theorems on subshifts
- What about theorems on groups ?