Sous-shifts, conjugaison et calculabilité

Pascal Vanier

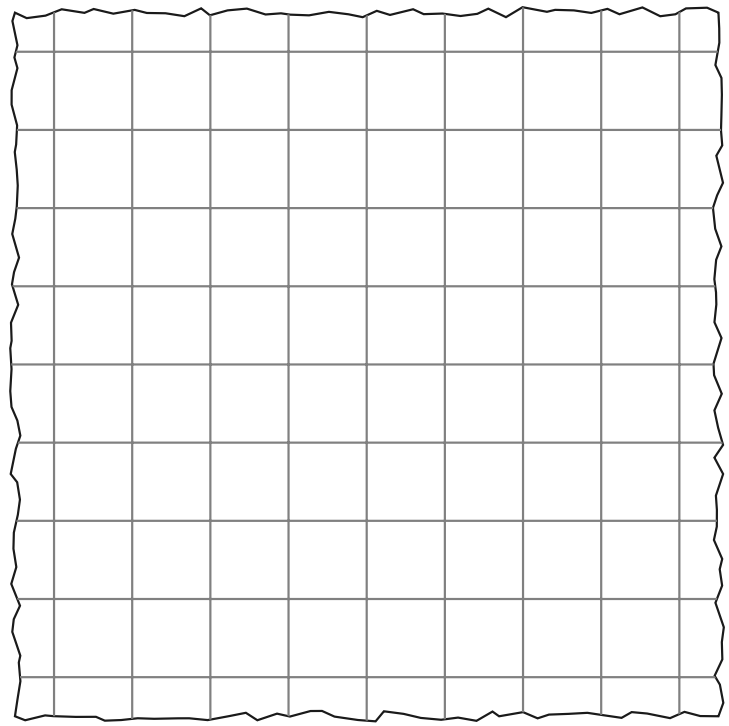
Laboratoire d'Algorithmique Complexité et Logique, UPEC

Séminaire LIPN

Sous-shifts de type fini et pavages

Un alphabet **fini** :

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

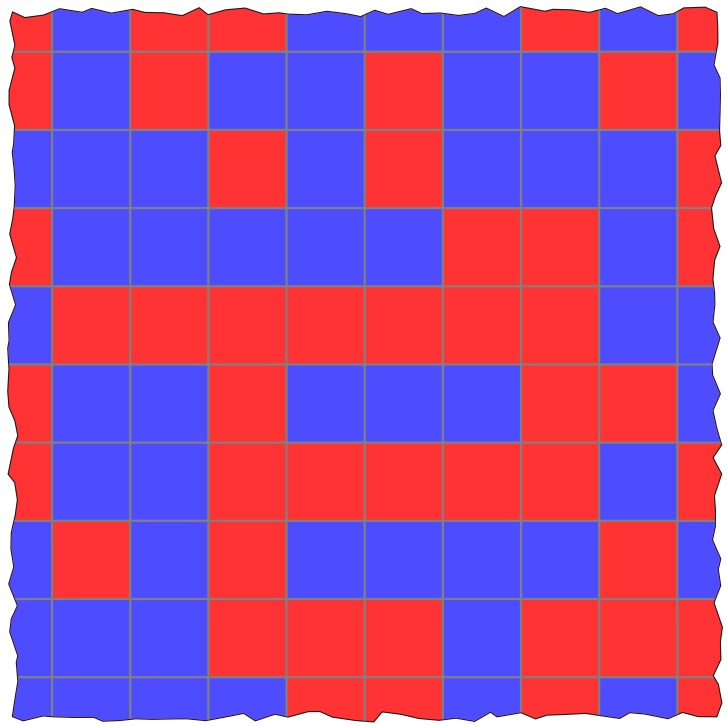


Sous-shifts de type fini et pavages

Un alphabet **fini** :

$\Sigma = \{$ ■, ■ $\}$

Un **pavage** ou **configuration** :



Sous-shifts de type fini et pavages

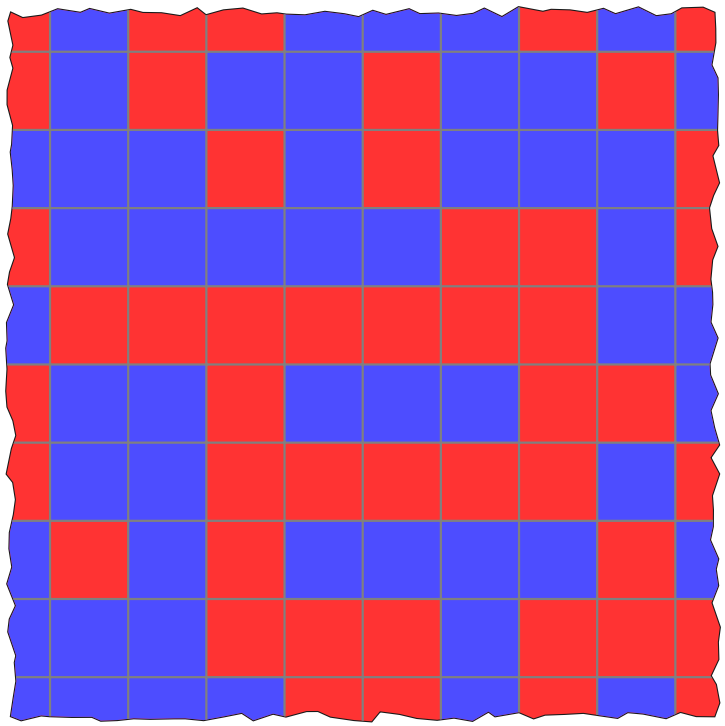
Un alphabet **fini** :

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

Des motifs interdits en nombre **fini** :

$$\mathcal{F} = \left\{ \blacksquare\blacksquare, \blacksquare\blacksquare, \begin{smallmatrix}\blacksquare\\\blacksquare\end{smallmatrix} \right\}$$

Un **pavage** ou **configuration** :



Sous-shifts de type fini et pavages

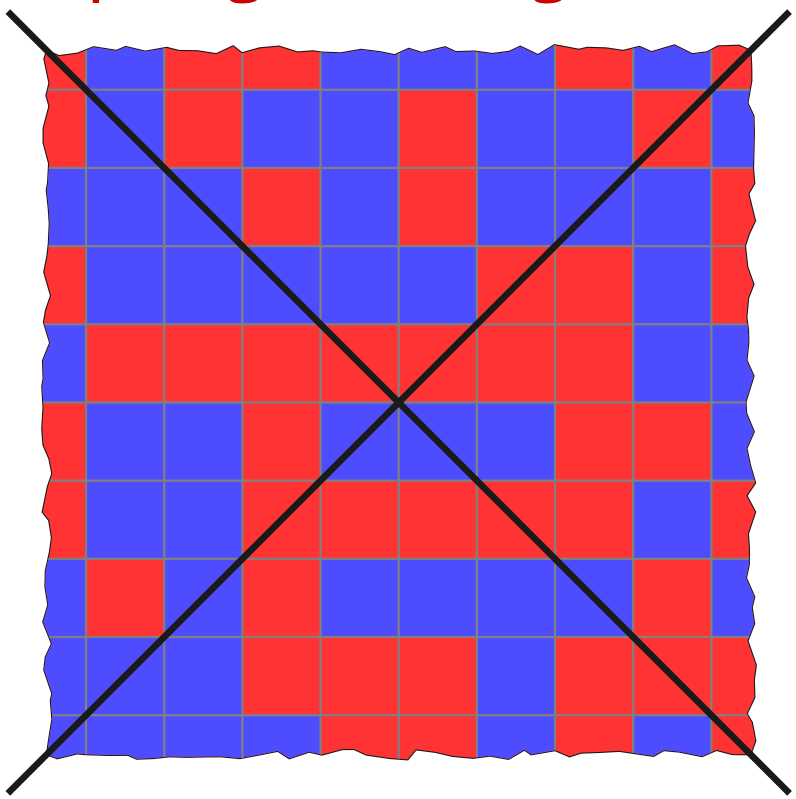
Un alphabet **fini** :

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

Des motifs interdits en nombre **fini** :

$$\mathcal{F} = \left\{ \blacksquare\blacksquare, \blacksquare\blacksquare, \begin{smallmatrix}\blacksquare\\\blacksquare\end{smallmatrix} \right\}$$

Un **pavage** ou **configuration** :



Sous-shifts de type fini et pavages

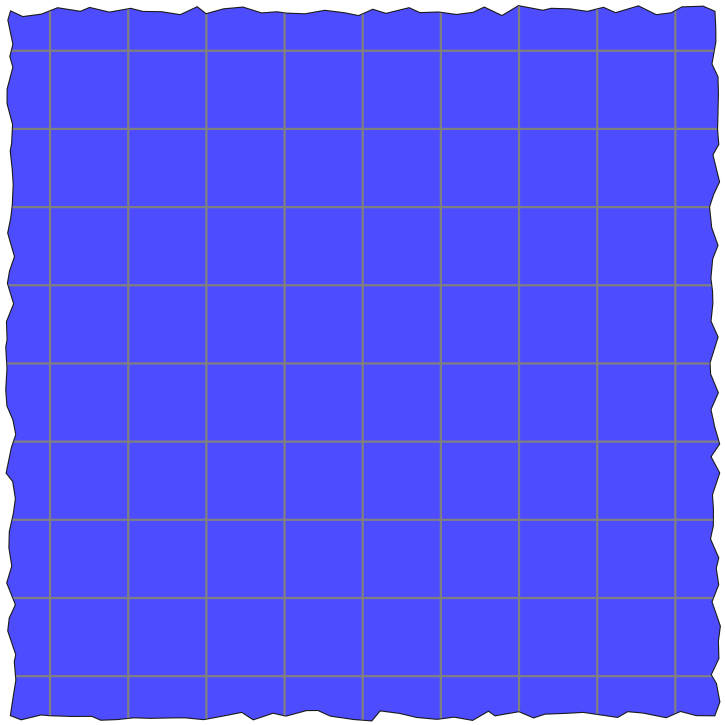
Un alphabet **fini** :

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

Des motifs interdits en nombre **fini** :

$$\mathcal{F} = \left\{ \square\square, \square\square, \begin{matrix}\square\\\square\end{matrix} \right\}$$

Un **pavage** ou **configuration** :



Sous-shifts de type fini et pavages

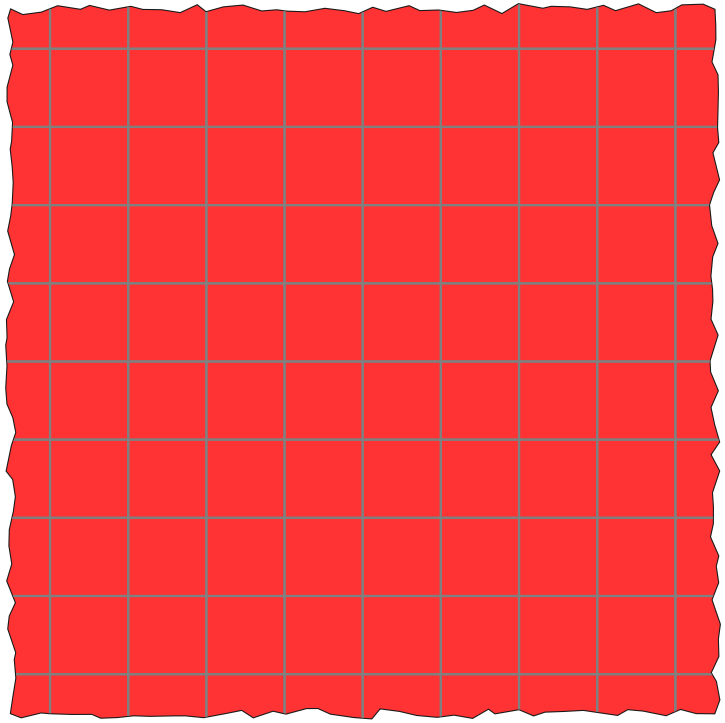
Un alphabet **fini** :

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

Des motifs interdits en nombre **fini** :

$$\mathcal{F} = \left\{ \square\square, \square\square, \begin{matrix}\square\\\square\end{matrix} \right\}$$

Un **pavage** ou **configuration** :



Sous-shifts de type fini et pavages

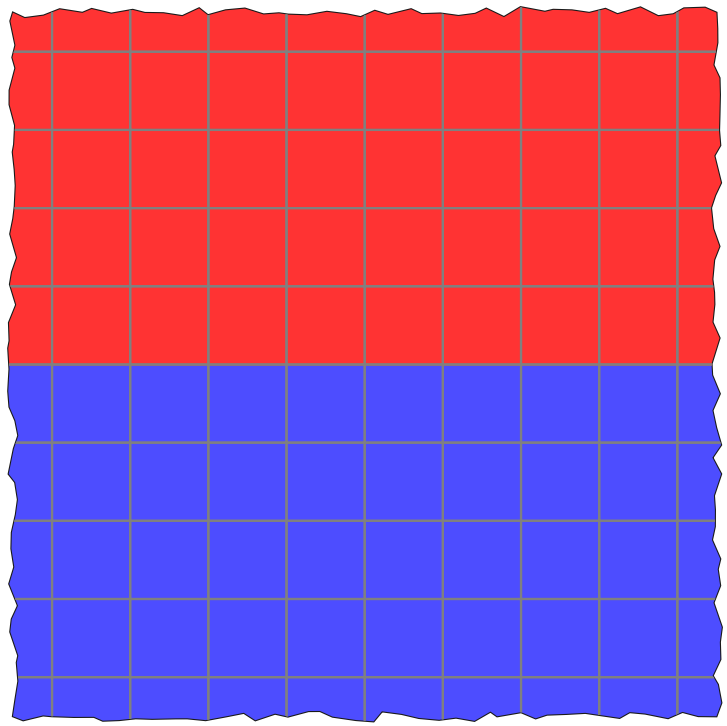
Un alphabet **fini** :

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

Des motifs interdits en nombre **fini** :

$$\mathcal{F} = \left\{ \square\square, \square\square, \begin{matrix}\square\\\square\end{matrix} \right\}$$

Un **pavage** ou **configuration** :



Sous-shifts de type fini et pavages

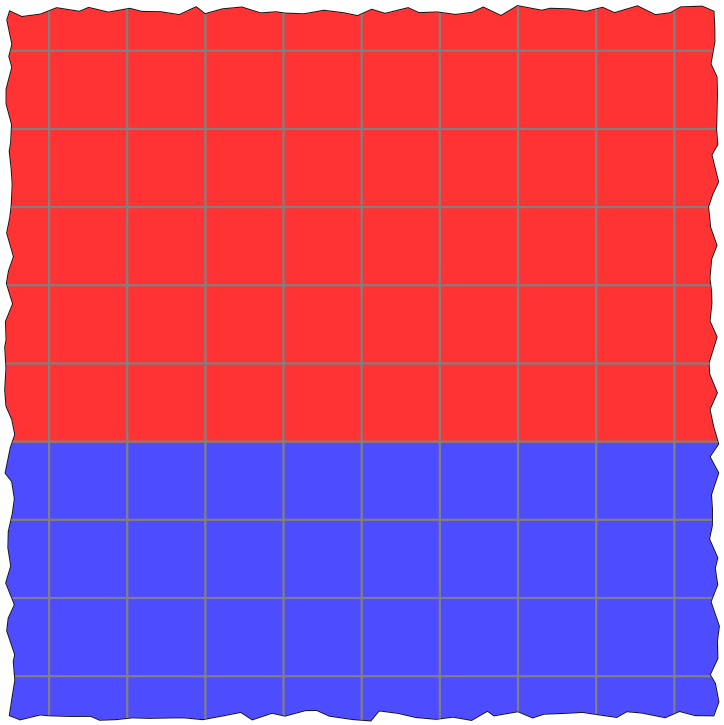
Un alphabet **fini** :

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

Des motifs interdits en nombre **fini** :

$$\mathcal{F} = \left\{ \square\square, \square\square, \begin{matrix}\square\\\square\end{matrix} \right\}$$

Un **pavage** ou **configuration** :



Sous-shifts de type fini et pavages

Un alphabet **fini** :

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

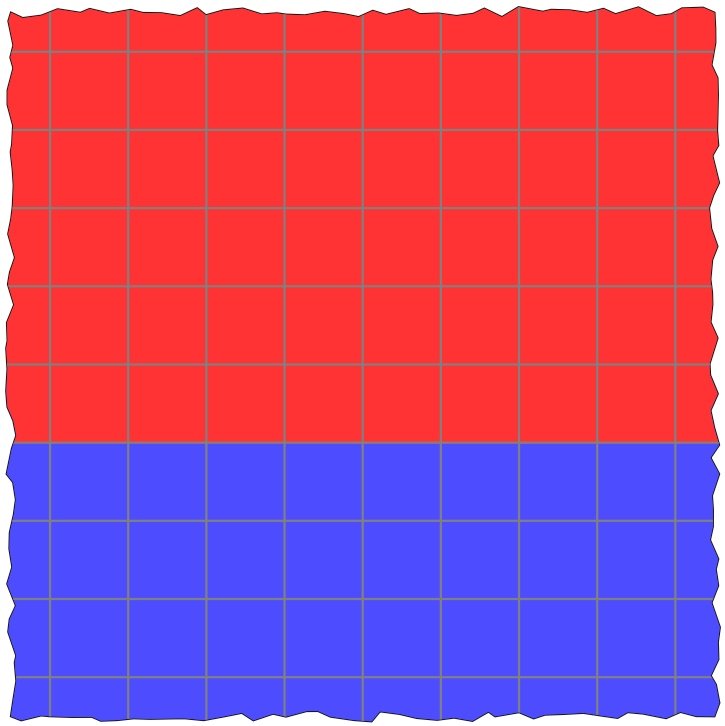
Des motifs interdits en nombre **fini** :

$$\mathcal{F} = \left\{ \square\square, \square\square, \begin{matrix}\square\\\square\end{matrix} \right\}$$

Sous-shift de Type Fini (SFT): l'ensemble configurations évitant $\mathcal{F}$. On note $\mathcal{X}_{\mathcal{F}}$:

$$\mathcal{X}_{\mathcal{F}} = \left\{ \square, \square, \square, \square, \ldots \right\}$$

Un **pavage** ou **configuration** :



Sous-shifts de type fini et pavages

Un alphabet **fini** :

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

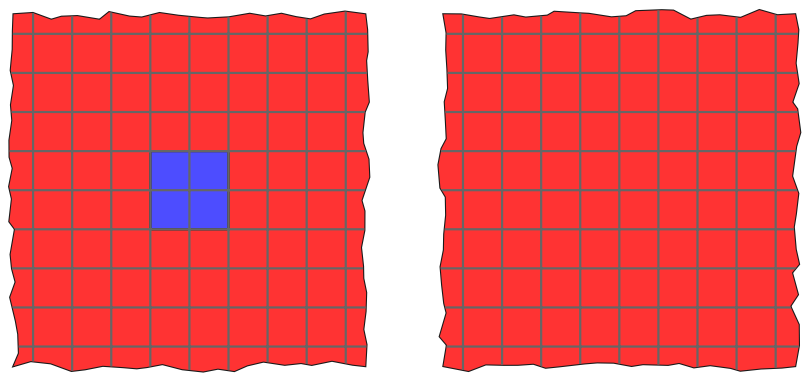
Des motifs interdits en nombre **fini** :

$$\mathcal{F} = \left\{ \blacksquare\blacksquare, \blacksquare\blacksquare, \begin{matrix}\blacksquare\\\blacksquare\end{matrix} \right\}$$

Sous-shift de Type Fini (SFT): l'ensemble configurations évitant $\mathcal{F}$. On note $\mathcal{X}_\mathcal{F}$:

$$\mathcal{X}_\mathcal{F} = \left\{ \blacksquare, \blacksquare, \begin{matrix}\blacksquare\\\blacksquare\end{matrix}, \begin{matrix}\blacksquare\\\blacksquare\end{matrix}, \ldots \right\}$$

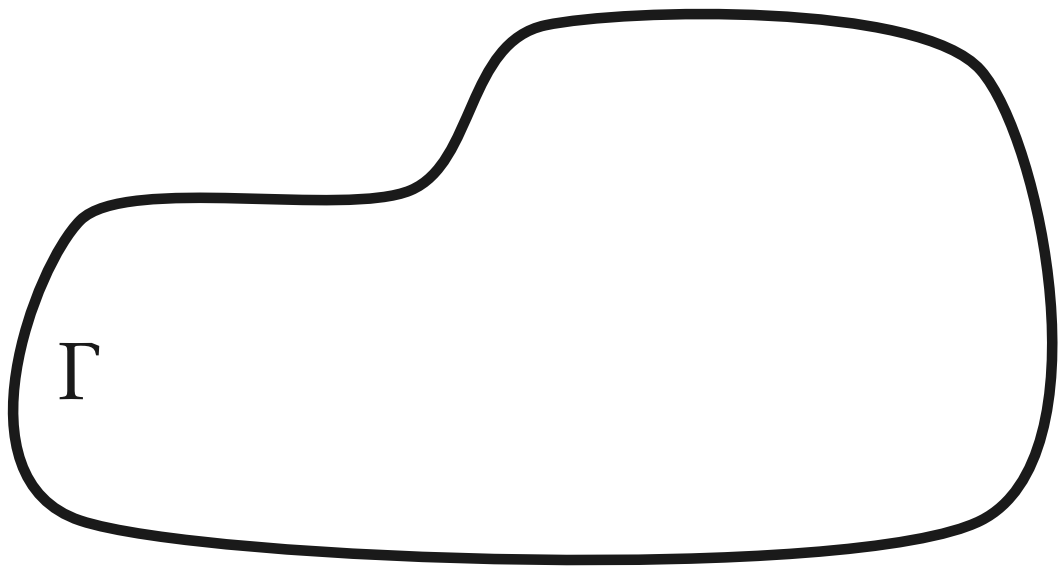
La **famille** peut également ne **pas** être **finie**, l'espace généré est alors un **sous-shift**.



Dynamique Symbolique : Origins

Les sous-shifts ont été introduits en dimension 1 :

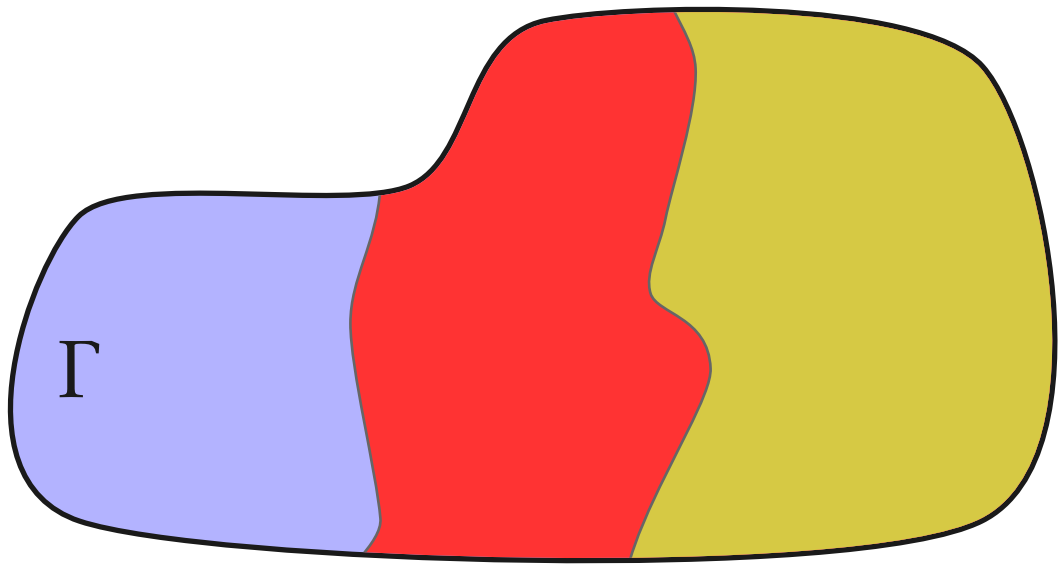
- $\phi : \Gamma \to \Gamma$ est un système dynamique.



Dynamique Symbolique : Origins

Les sous-shifts ont été introduits en dimension 1 :

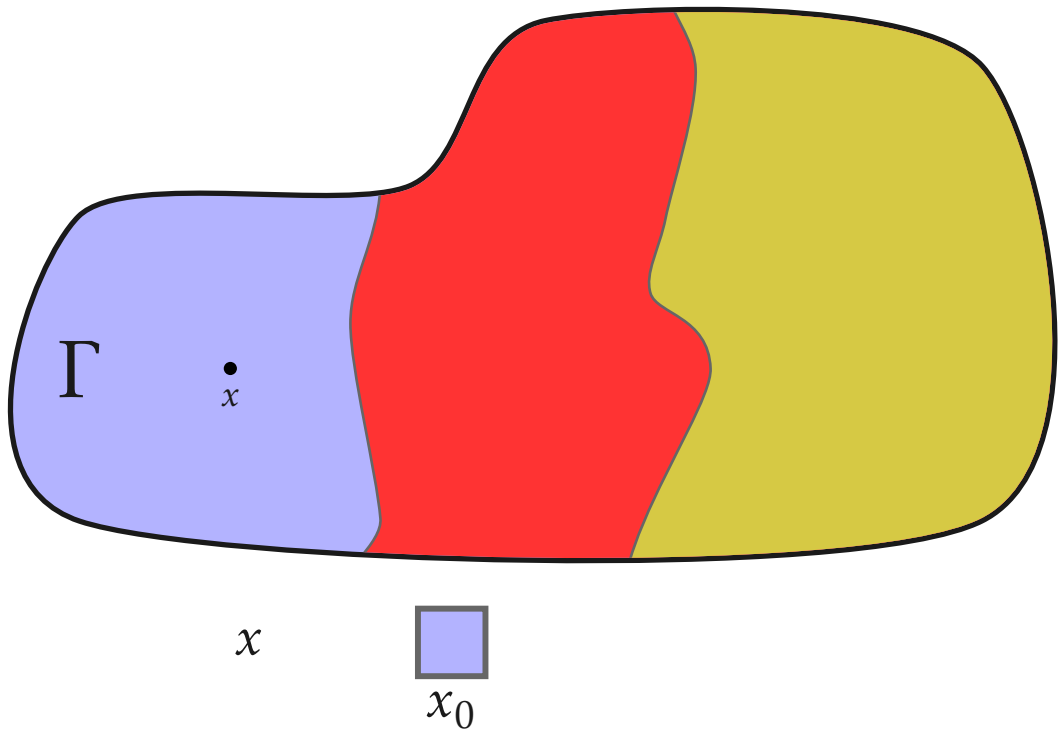
- $\phi : \Gamma \to \Gamma$ est un système dynamique.
- On partitionne Γ.



Dynamique Symbolique : Origins

Les sous-shifts ont été introduits en dimension 1 :

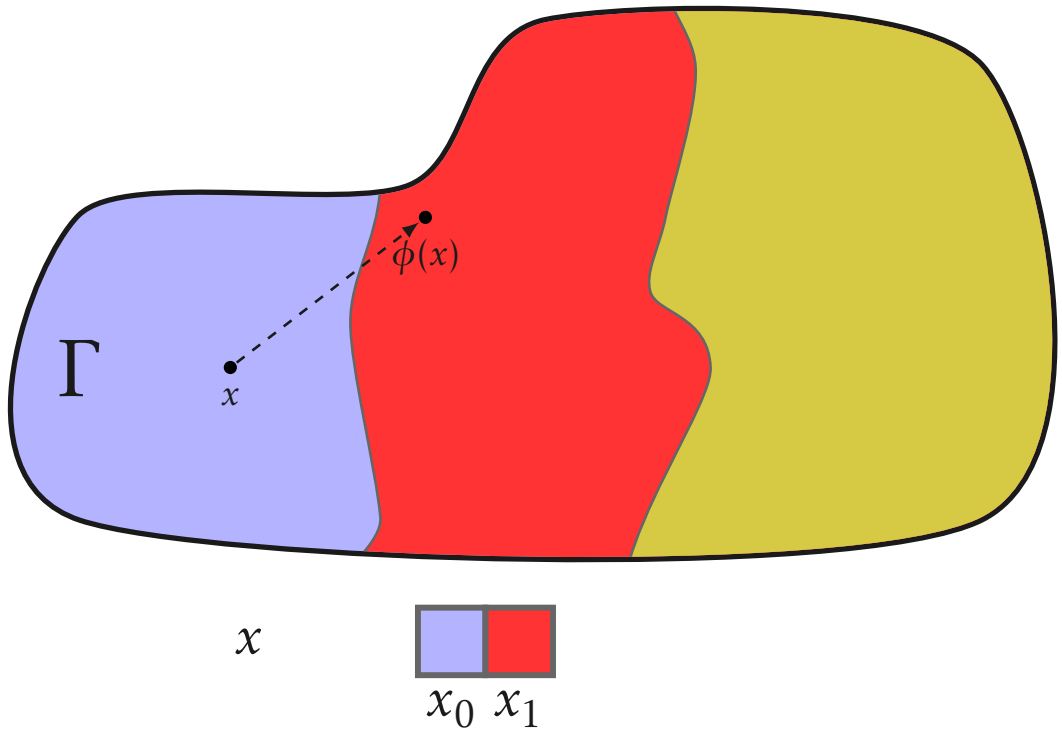
- $\phi : \Gamma \to \Gamma$ est un système dynamique.
- On partitionne Γ.
- Pour chaque point d'une orbite, on ne garde que la partition où il se trouve.



Dynamique Symbolique : Origins

Les sous-shifts ont été introduits en dimension 1 :

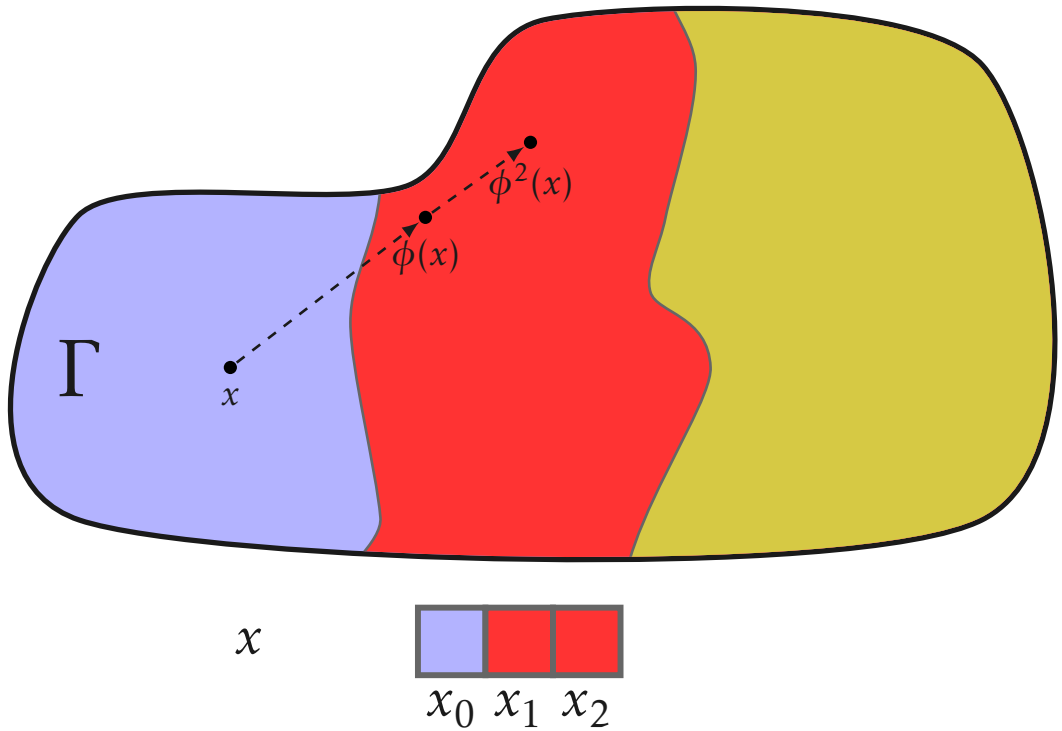
- $\phi : \Gamma \to \Gamma$ est un système dynamique.
- On partitionne Γ.
- Pour chaque point d'une orbite, on ne garde que la partition où il se trouve.



Dynamique Symbolique : Origins

Les sous-shifts ont été introduits en dimension 1 :

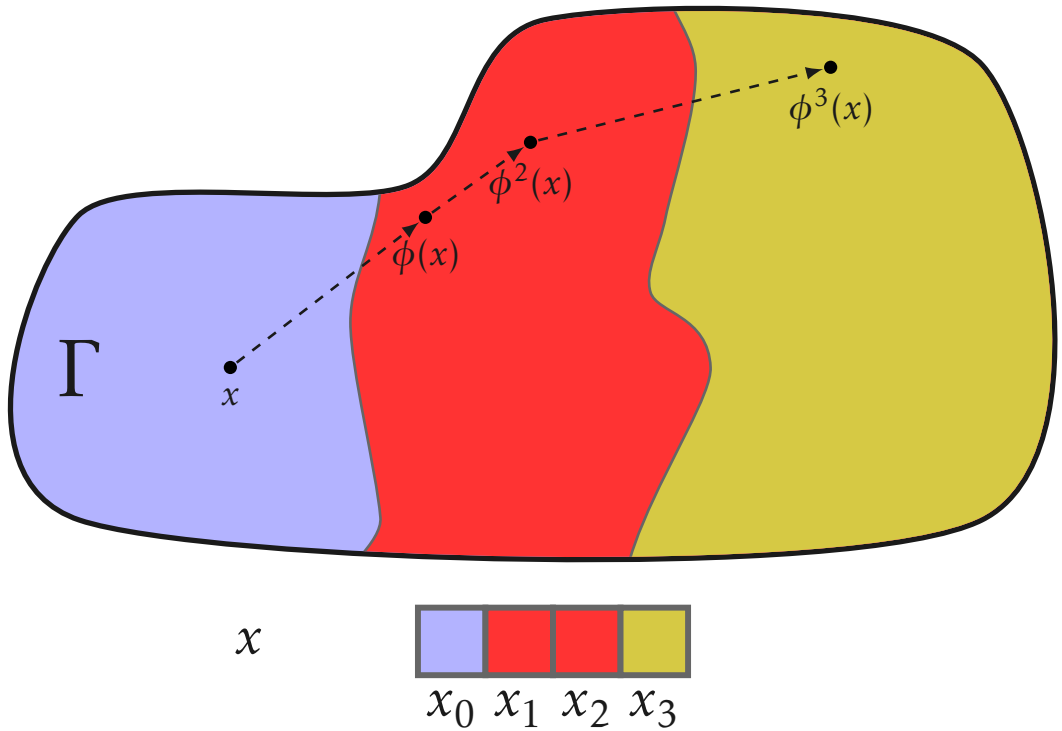
- $\phi : \Gamma \to \Gamma$ est un système dynamique.
- On partitionne Γ.
- Pour chaque point d'une orbite, on ne garde que la partition où il se trouve.



Dynamique Symbolique : Origins

Les sous-shifts ont été introduits en dimension 1 :

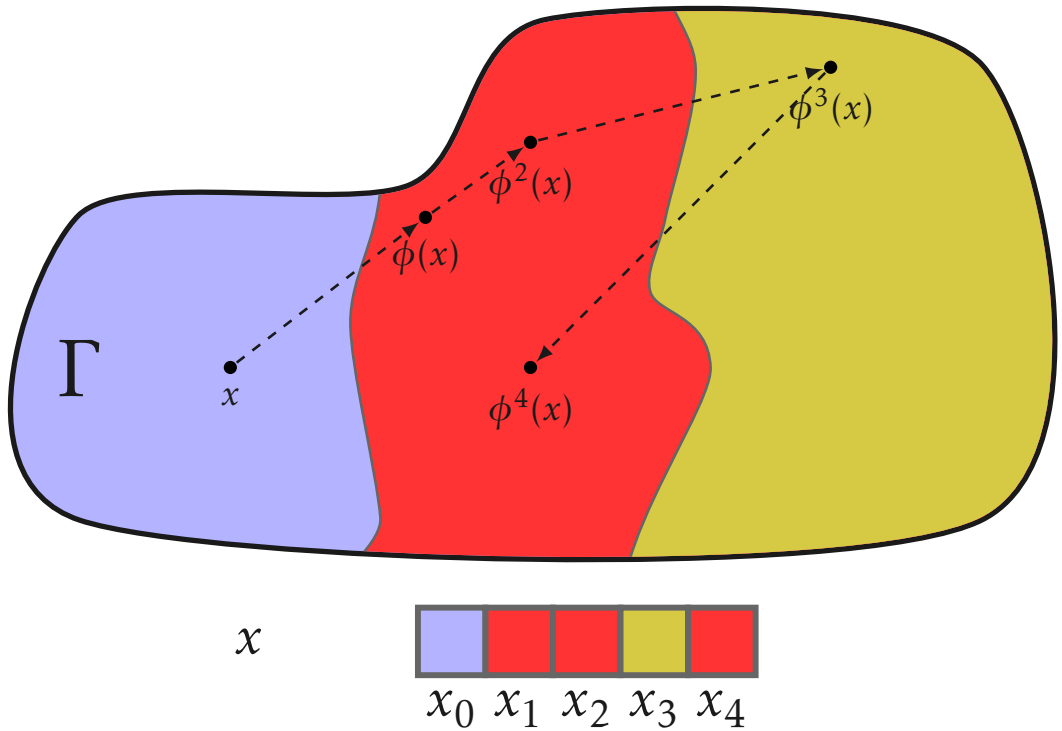
- $\phi : \Gamma \to \Gamma$ est un système dynamique.
- On partitionne Γ.
- Pour chaque point d'une orbite, on ne garde que la partition où il se trouve.



Dynamique Symbolique : Origins

Les sous-shifts ont été introduits en dimension 1 :

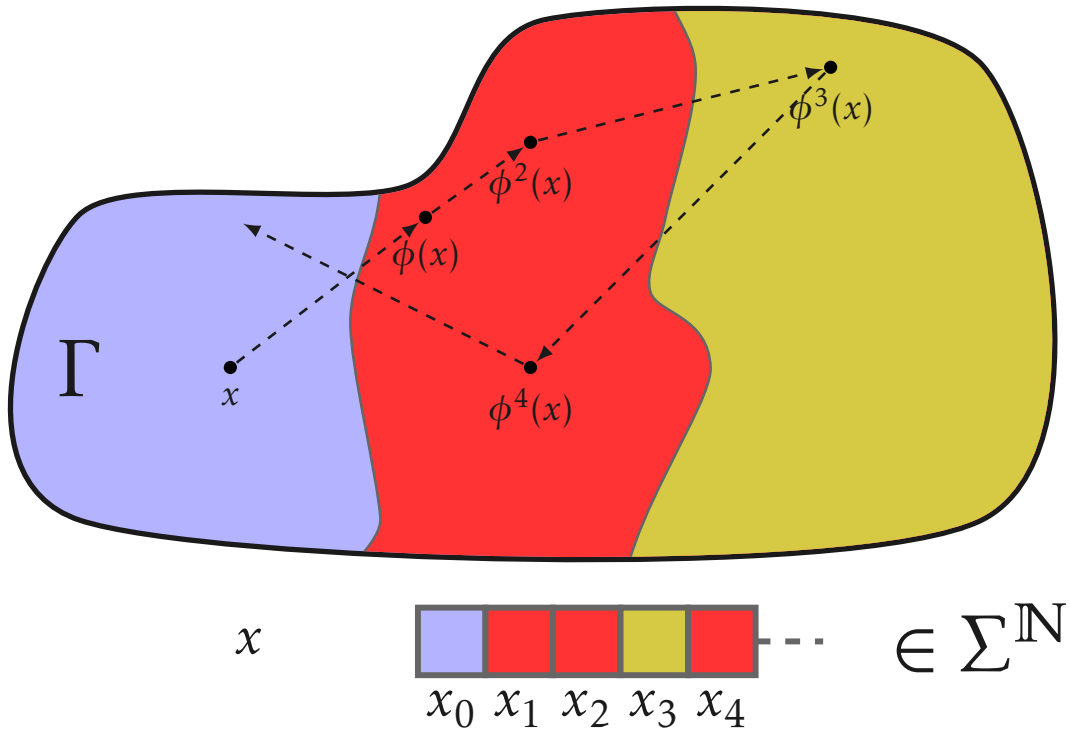
- $\phi : \Gamma \to \Gamma$ est un système dynamique.
- On partitionne Γ.
- Pour chaque point d'une orbite, on ne garde que la partition où il se trouve.



Dynamique Symbolique : Origins

Les sous-shifts ont été introduits en dimension 1 :

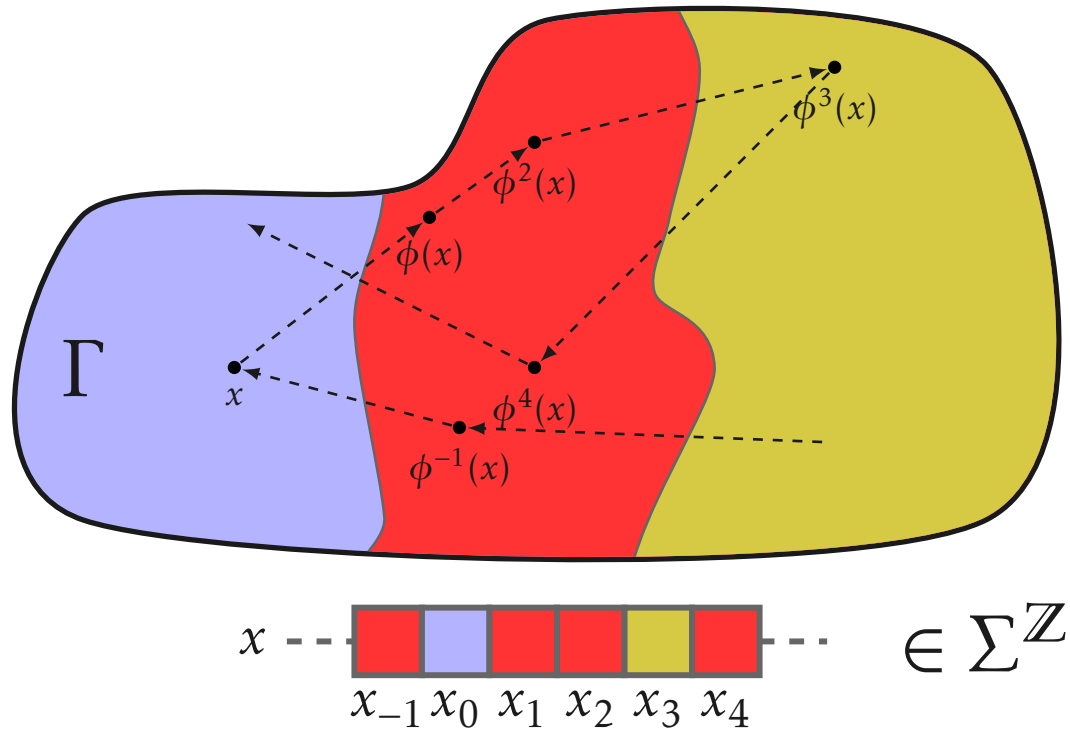
- $\phi : \Gamma \to \Gamma$ est un système dynamique.
- On partitionne Γ.
- Pour chaque point d'une orbite, on ne garde que la partition où il se trouve.



Dynamique Symbolique : Origins

Les sous-shifts ont été introduits en dimension 1 :

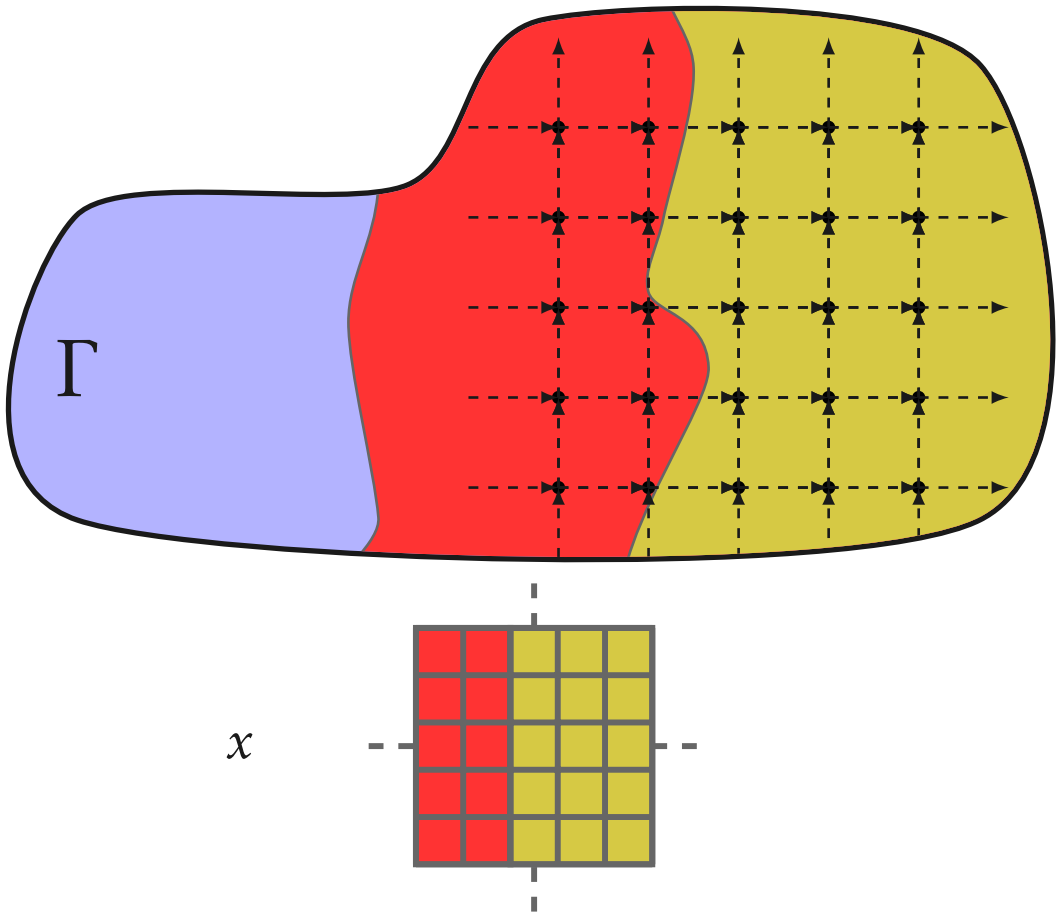
- $\phi : \Gamma \to \Gamma$ est un système dynamique.
- On partitionne Γ.
- Pour chaque point d'une orbite, on ne garde que la partition où il se trouve.



Dynamique Symbolique : Origins

Les sous-shifts en dimension 2 :

- $\phi_1, \phi_2 : \Gamma \to \Gamma$, $\phi_1 \circ \phi_2 = \phi_2 \circ \phi_1$ est un système dynamique.
- On partitionne Γ.
- Pour chaque point d'une orbite, on ne garde que la partition où il se trouve.



Dynamique Symbolique 1d

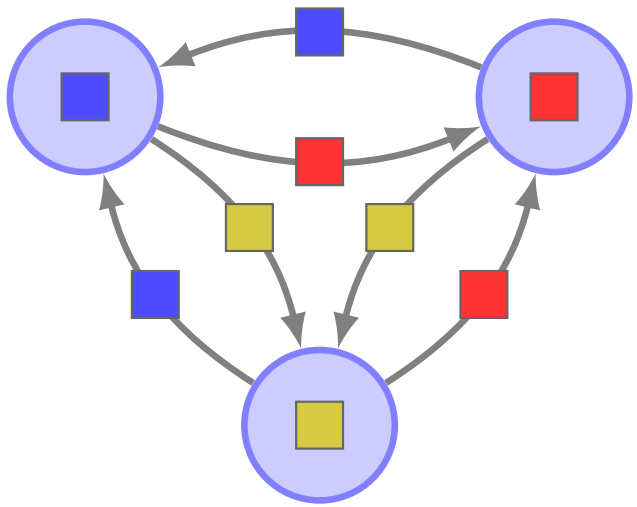
En dimension 1, les SFTs sont bien compris :

Un SFT de dimension 1 peut être représenté par un graphe. Les points sont alors les marches biinfinies sur ce graphe.

Par exemple :

$\Sigma = \{$  $\}$

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



Savoir si un SFT est non-vide revient à trouver un cycle dans ce graphe.

Un SFT non-vide contient donc toujours un point périodique.

Dynamique Symbolique 1d

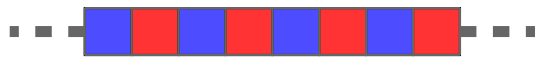
En dimension 1, les SFTs sont bien compris :

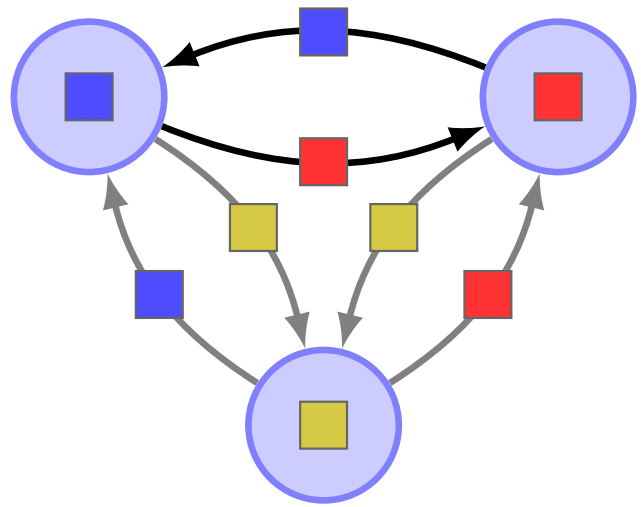
Un SFT de dimension 1 peut être représenté par un graphe. Les points sont alors les marches biinfinies sur ce graphe.

Par exemple :

$\Sigma =$ {■, ■, ■}

$\mathcal{F} =$ {■■, ■■, ■■}





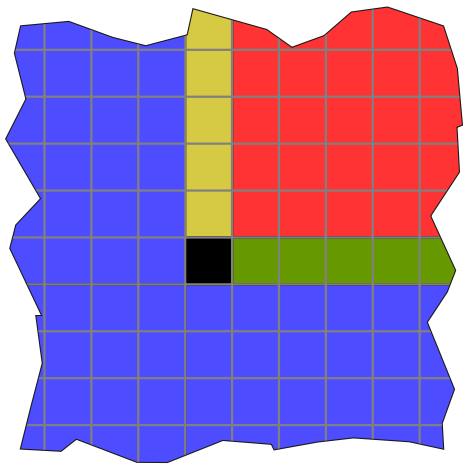
Savoir si un SFT est non-vide revient à trouver un cycle dans ce graphe.

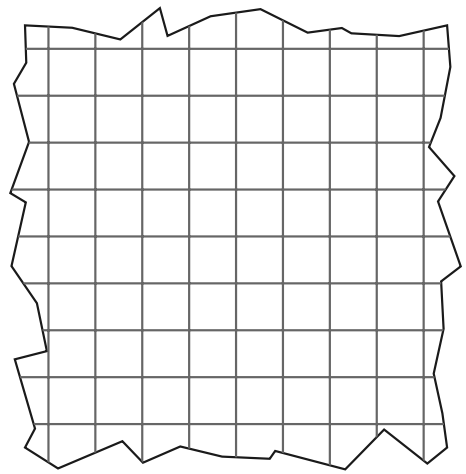
Un SFT non-vide contient donc toujours un point périodique.

Sous-shifts sofiques

Les graphes permettent de représenter plus que les sous-shifts de type fini.

Les **sous-shifts sofiques** sont les **projections** de SFTs par une **fonction locale**.



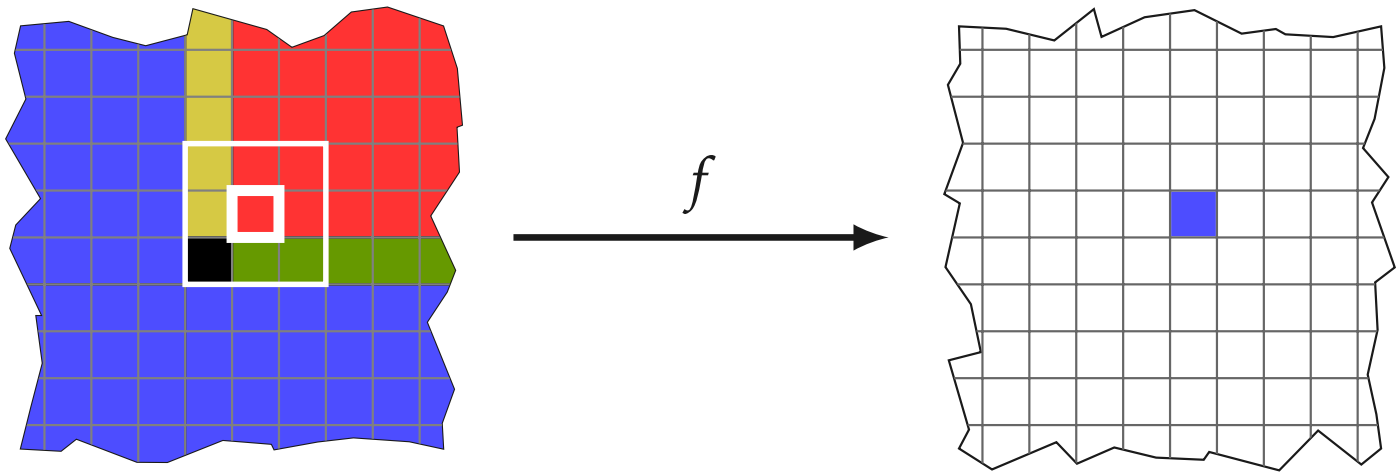


Remarque Les projections locales sont appelées **fonctions de factorisation**.

Sous-shifts sofiques

Les graphes permettent de représenter plus que les sous-shifts de type fini.

Les **sous-shifts sofiques** sont les **projections** de SFTs par une **fonction locale**.

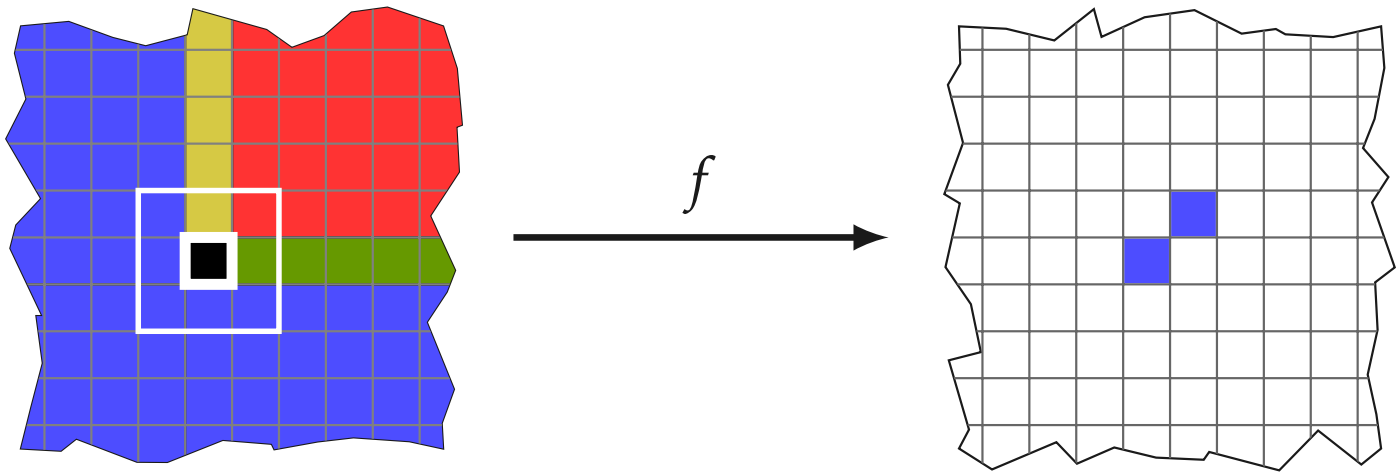


Remarque Les projections locales sont appelées **fonctions de factorisation**.

Sous-shifts sofiques

Les graphes permettent de représenter plus que les sous-shifts de type fini.

Les **sous-shifts sofiques** sont les **projections** de SFTs par une **fonction locale**.

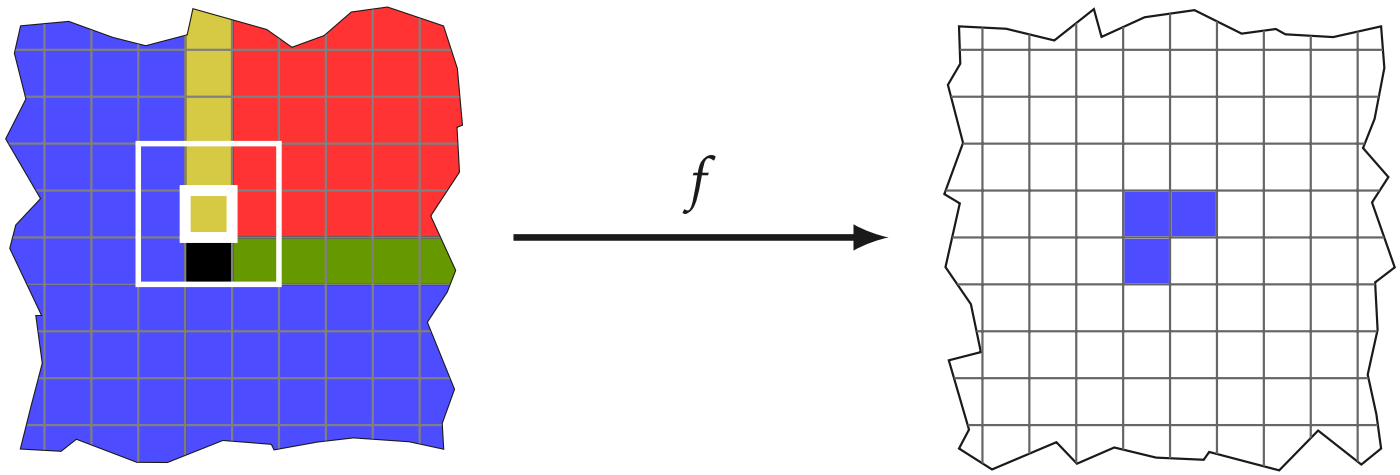


Remarque Les projections locales sont appelées **fonctions de factorisation**.

Sous-shifts sofiques

Les graphes permettent de représenter plus que les sous-shifts de type fini.

Les **sous-shifts sofiques** sont les **projections** de SFTs par une **fonction locale**.

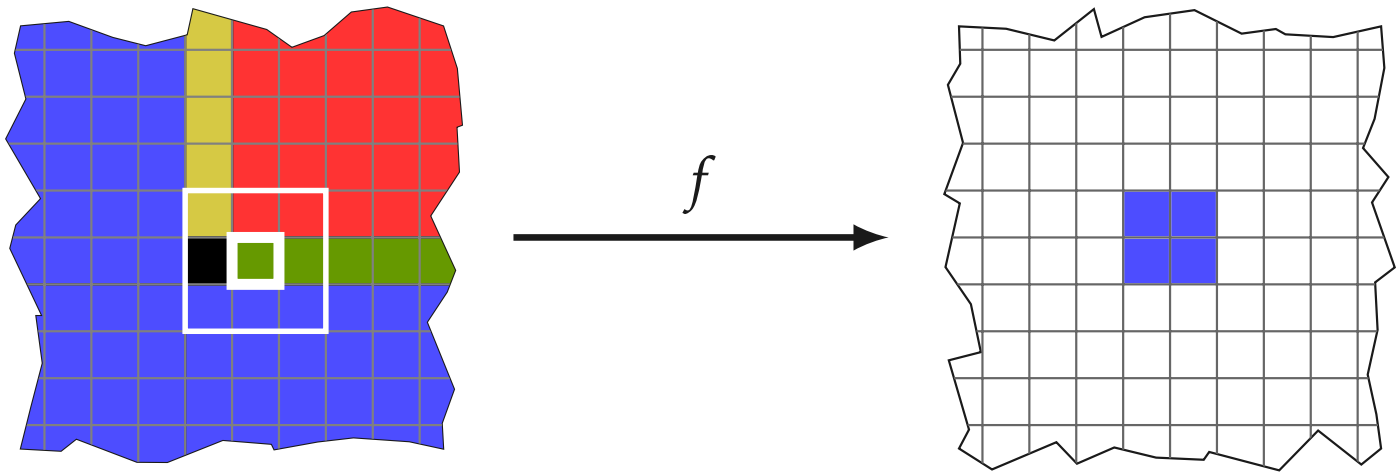


Remarque Les projections locales sont appelées **fonctions de factorisation**.

Sous-shifts sofiques

Les graphes permettent de représenter plus que les sous-shifts de type fini.

Les **sous-shifts sofiques** sont les **projections** de SFTs par une **fonction locale**.

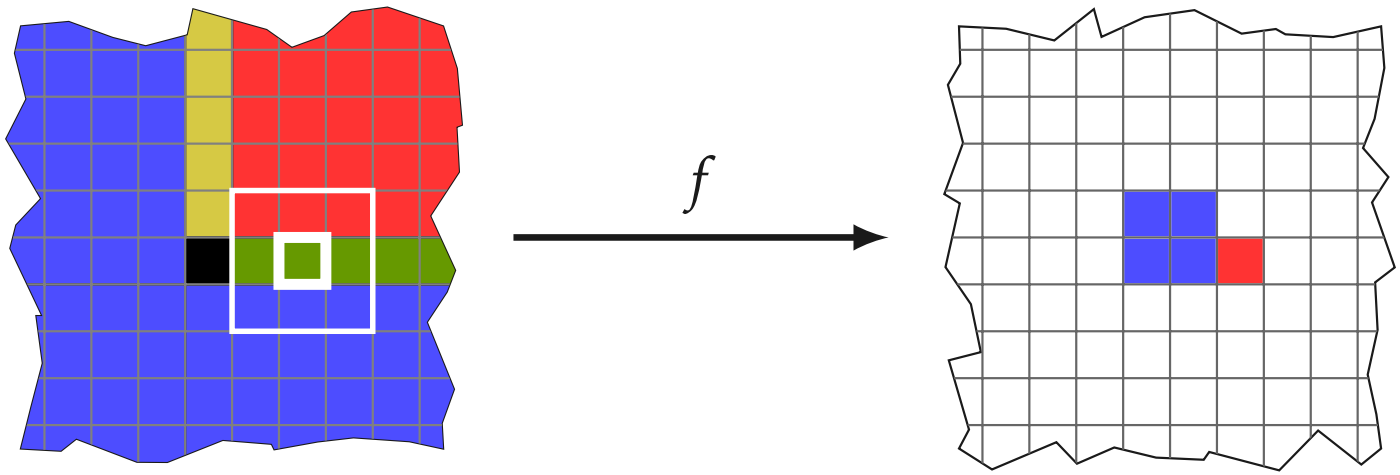


Remarque Les projections locales sont appelées **fonctions de factorisation**.

Sous-shifts sofiques

Les graphes permettent de représenter plus que les sous-shifts de type fini.

Les **sous-shifts sofiques** sont les **projections** de SFTs par une **fonction locale**.

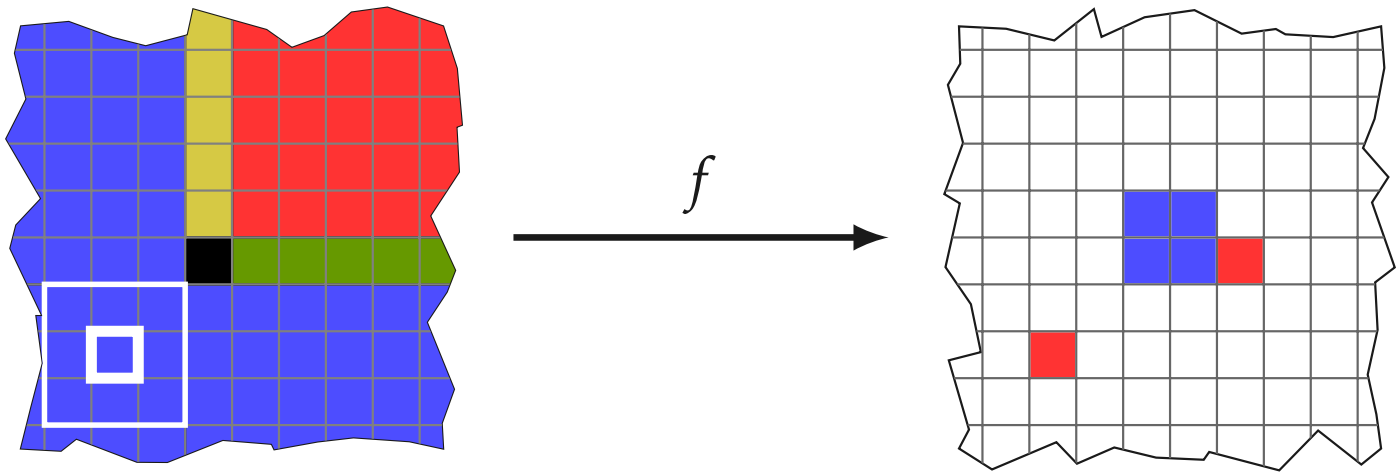


Remarque Les projections locales sont appelées **fonctions de factorisation**.

Sous-shifts sofiques

Les graphes permettent de représenter plus que les sous-shifts de type fini.

Les **sous-shifts sofiques** sont les **projections** de SFTs par une **fonction locale**.

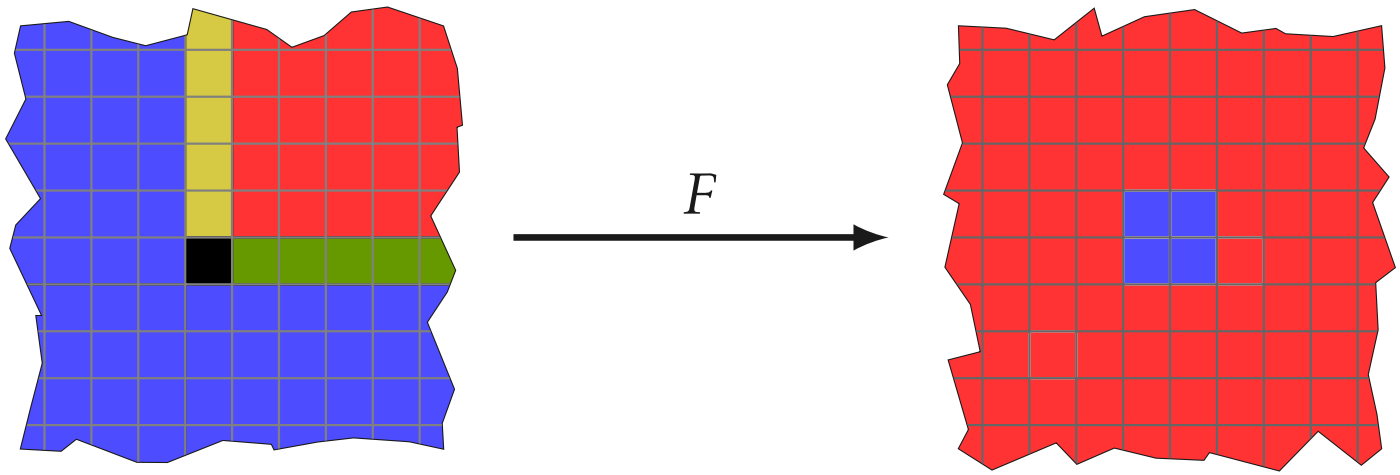


Remarque Les projections locales sont appelées **fonctions de factorisation**.

Sous-shifts sofiques

Les graphes permettent de représenter plus que les sous-shifts de type fini.

Les **sous-shifts sofiques** sont les **projections** de SFTs par une **fonction locale**.



Remarque Les projections locales sont appelées **fonctions de factorisation**.

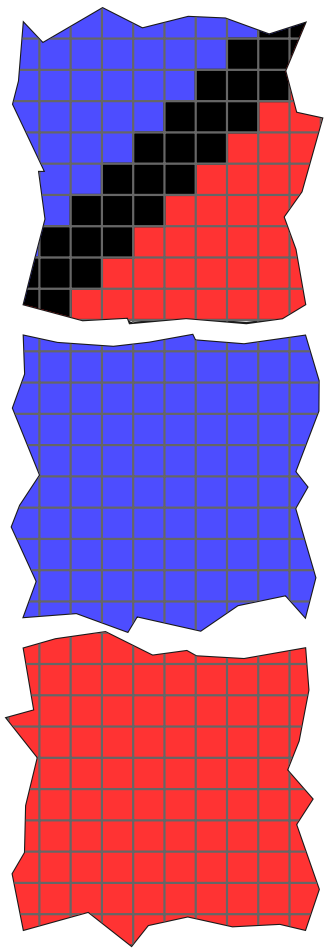
Conjugaison

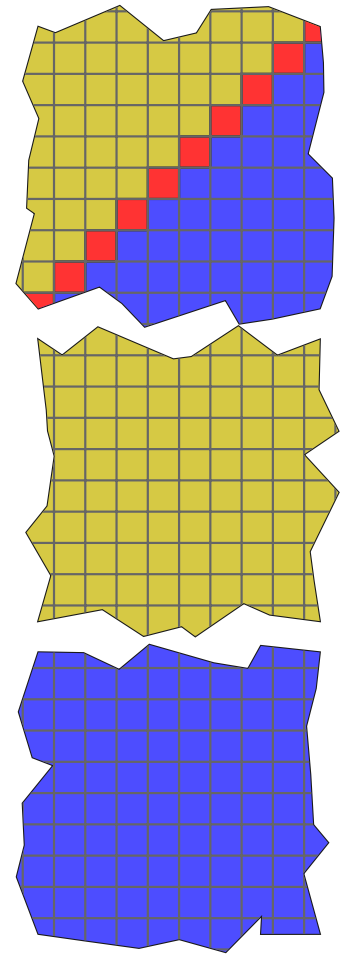
Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

Les deux sous-shifts suivants par exemple :

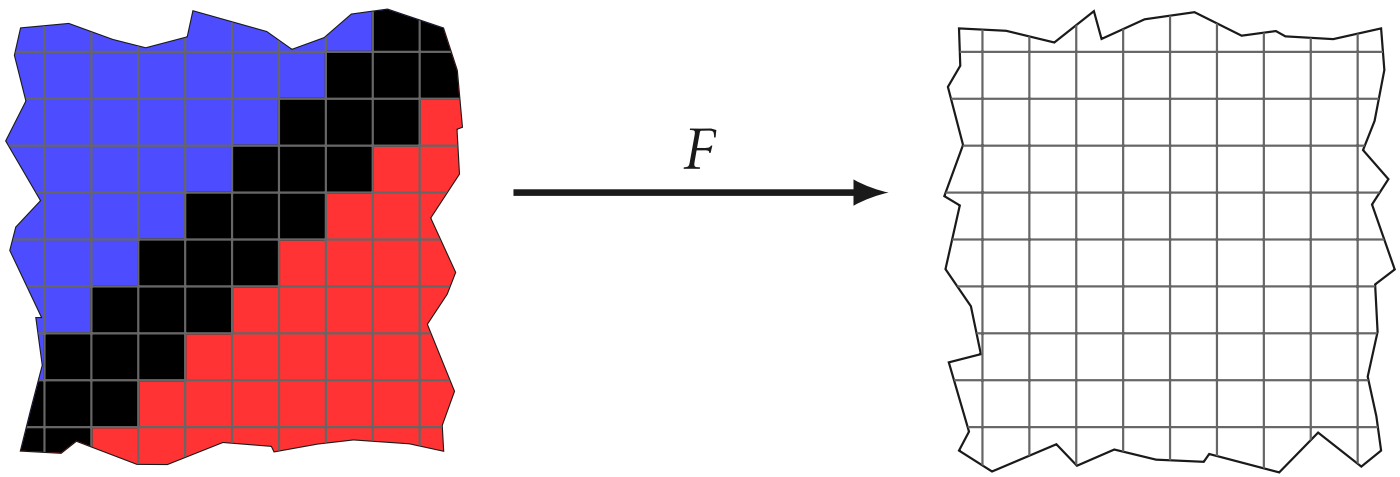




Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

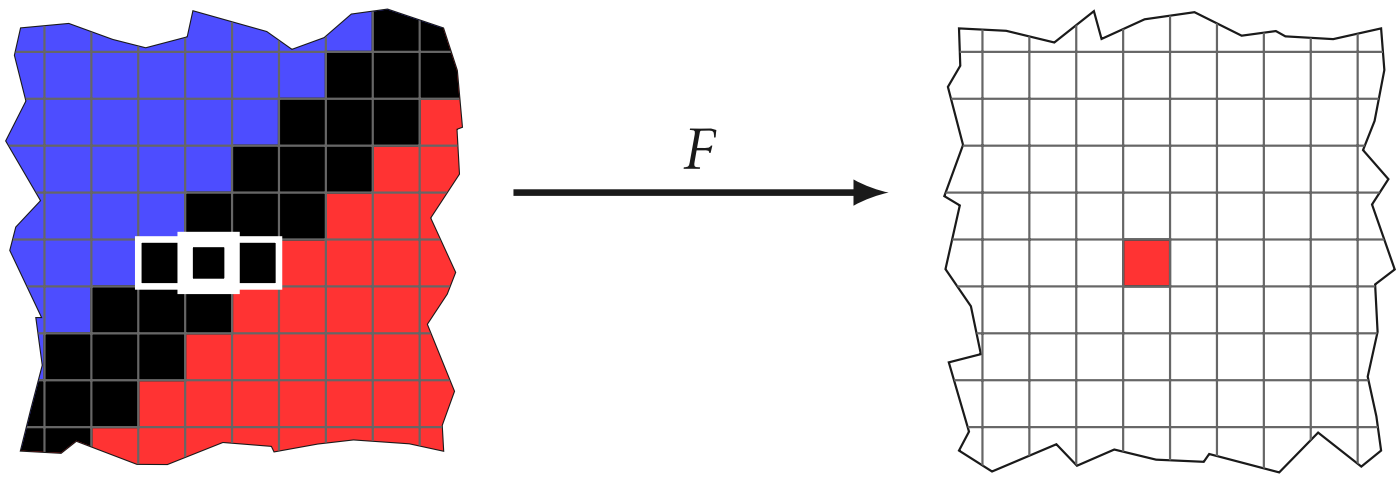
Une fonction de **conjugaison** est une projection locale **réversible** localement.



Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

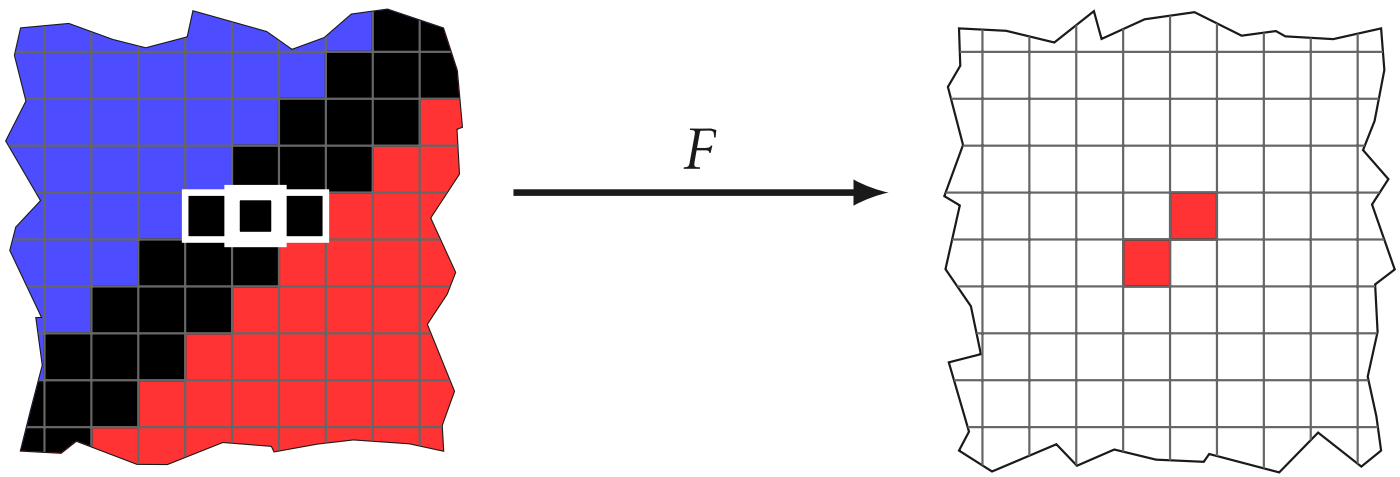
Une fonction de **conjugaison** est une projection locale **réversible** localement.



Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

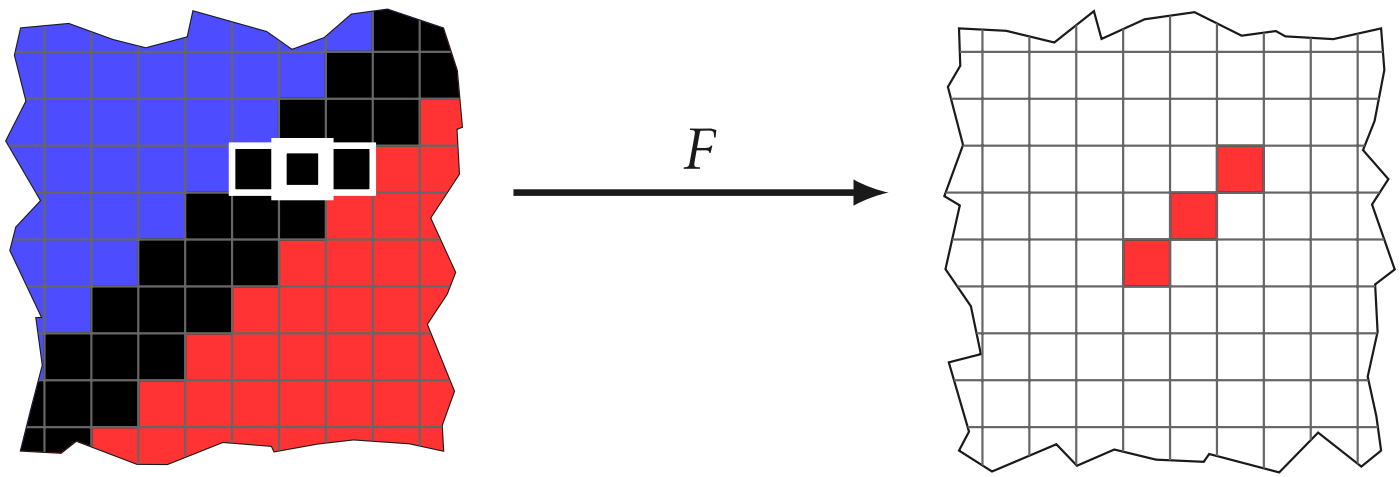
Une fonction de **conjugaison** est une projection locale **réversible** localement.



Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

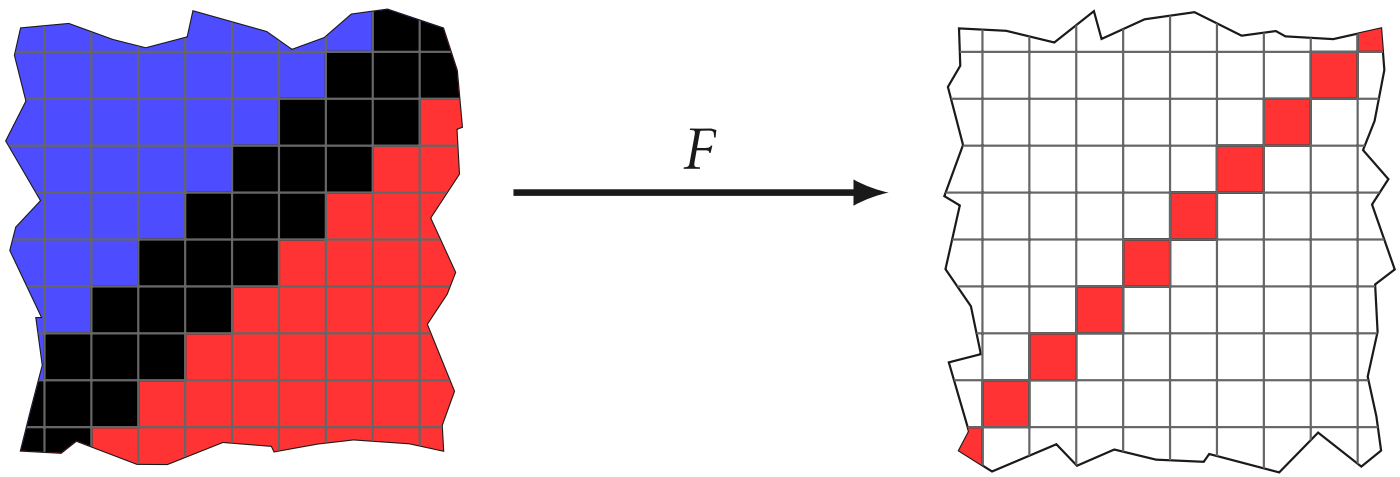
Une fonction de **conjugaison** est une projection locale **réversible** localement.



Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

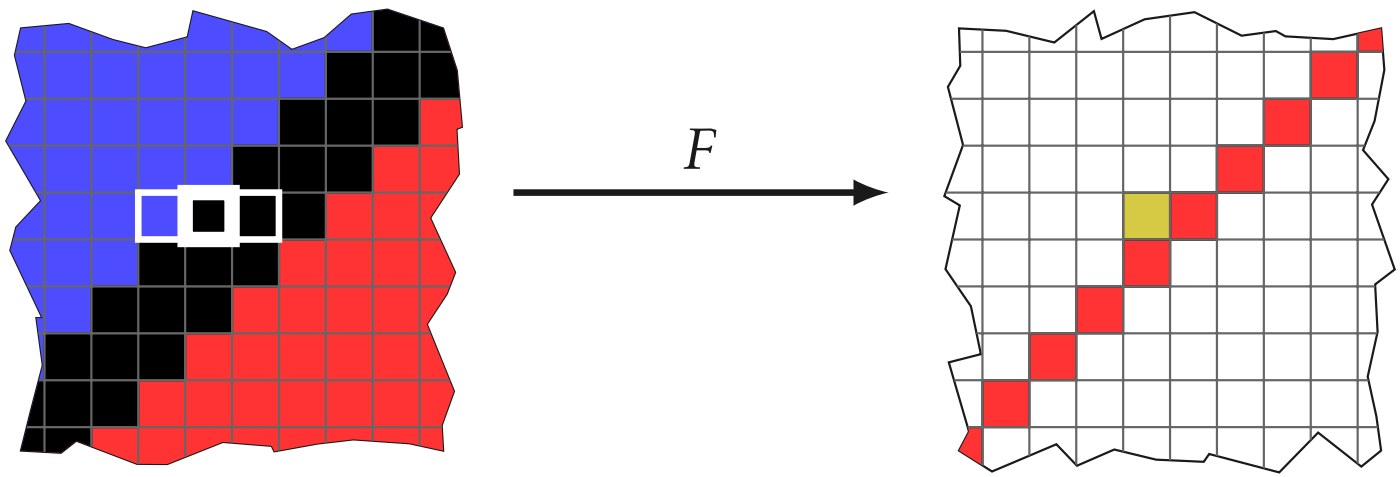
Une fonction de **conjugaison** est une projection locale **réversible** localement.



Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

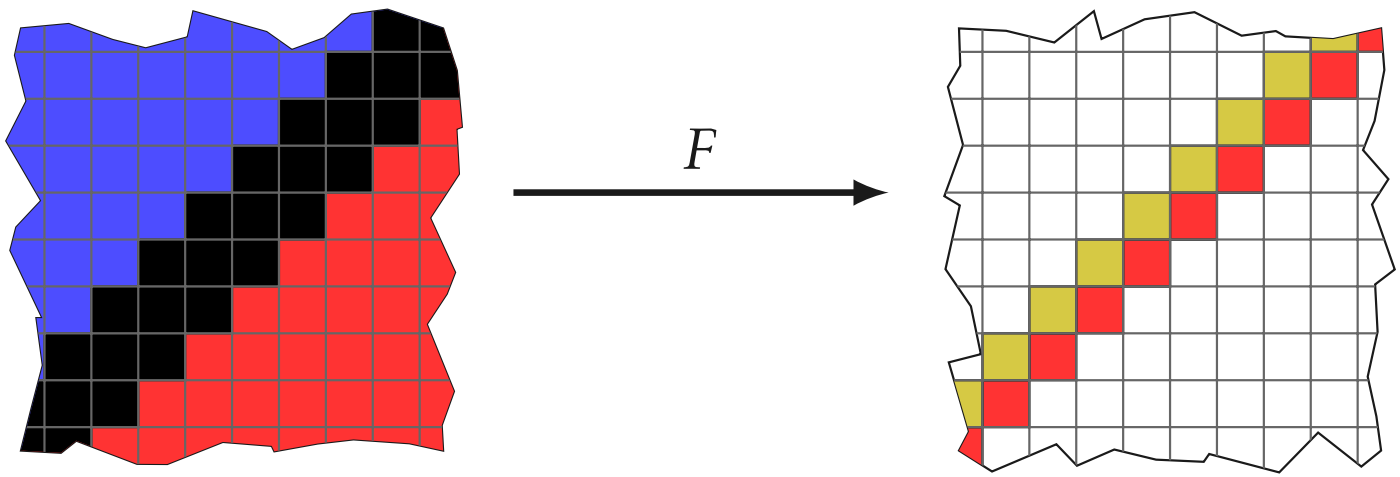
Une fonction de **conjugaison** est une projection locale **réversible** localement.



Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

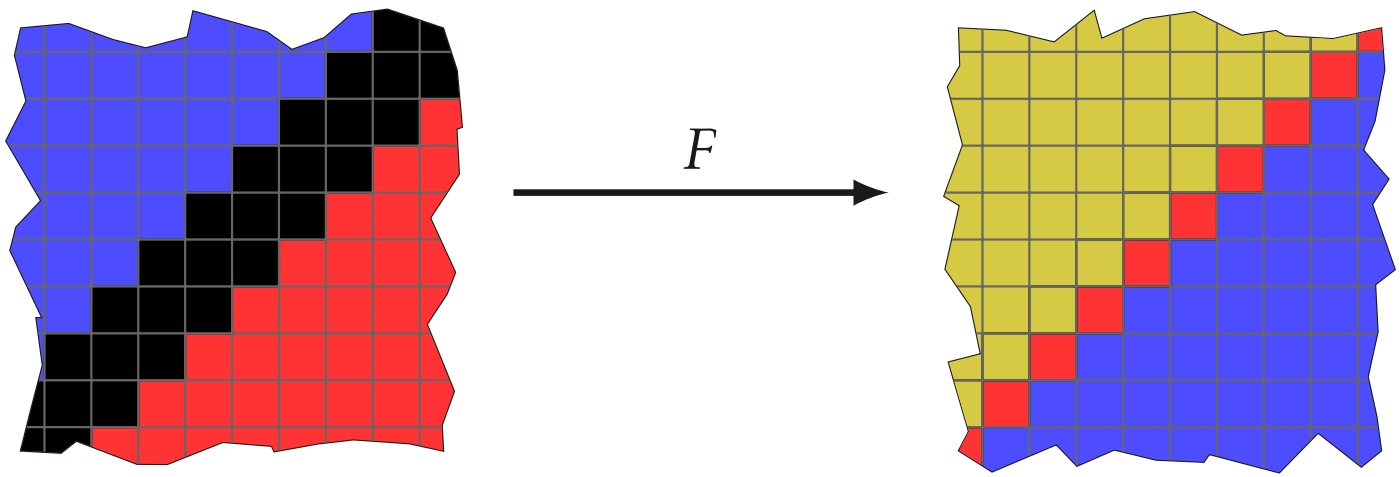
Une fonction de **conjugaison** est une projection locale **réversible** localement.



Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

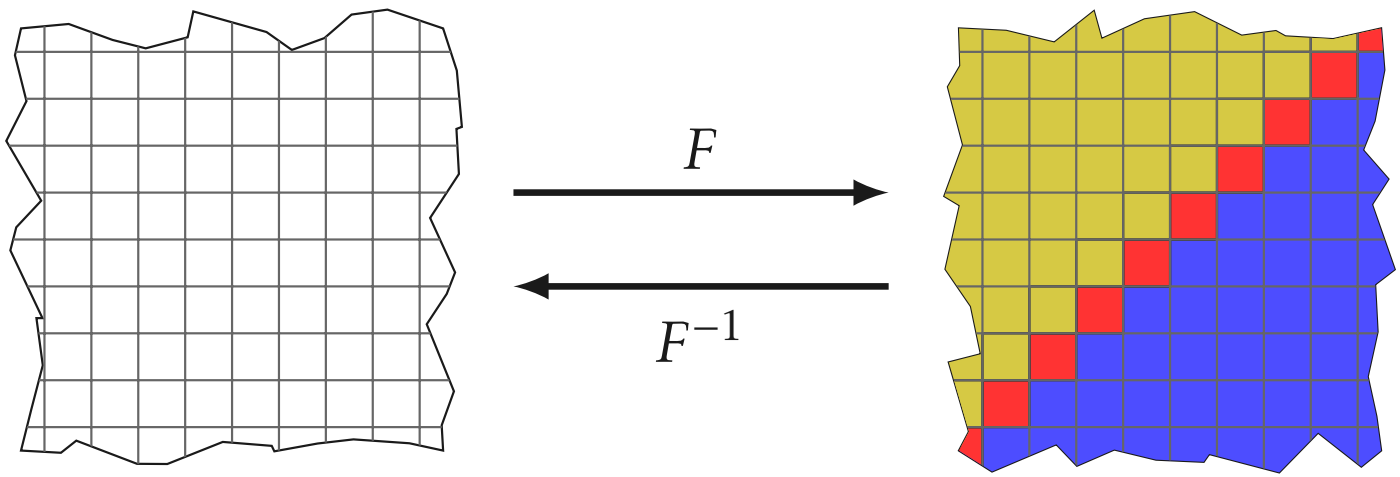
Une fonction de **conjugaison** est une projection locale **réversible** localement.



Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

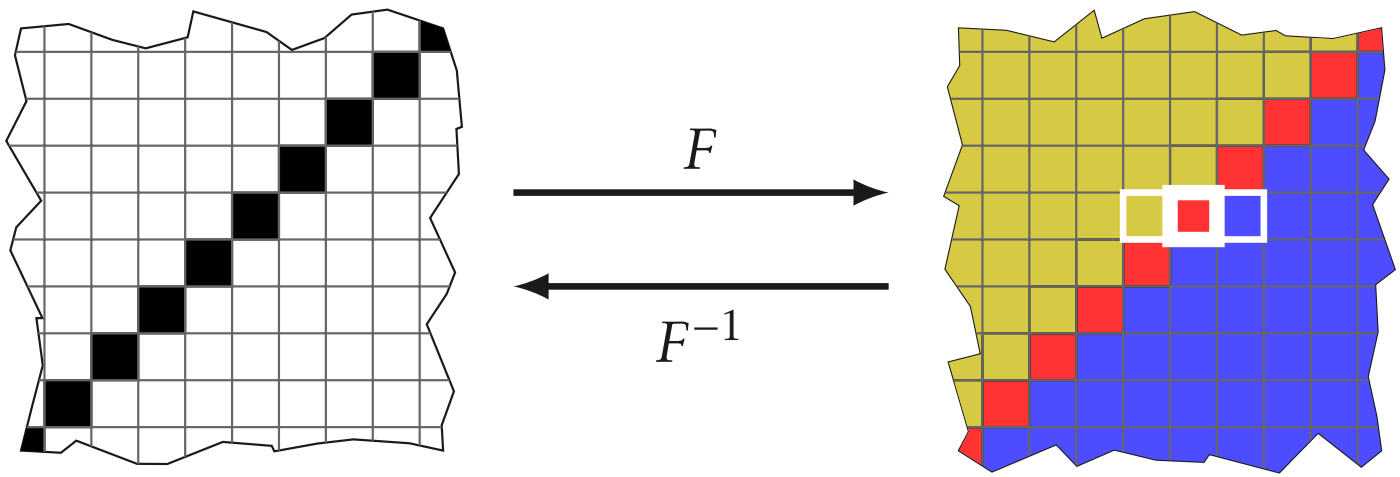
Une fonction de **conjugaison** est une projection locale **réversible** localement.



Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

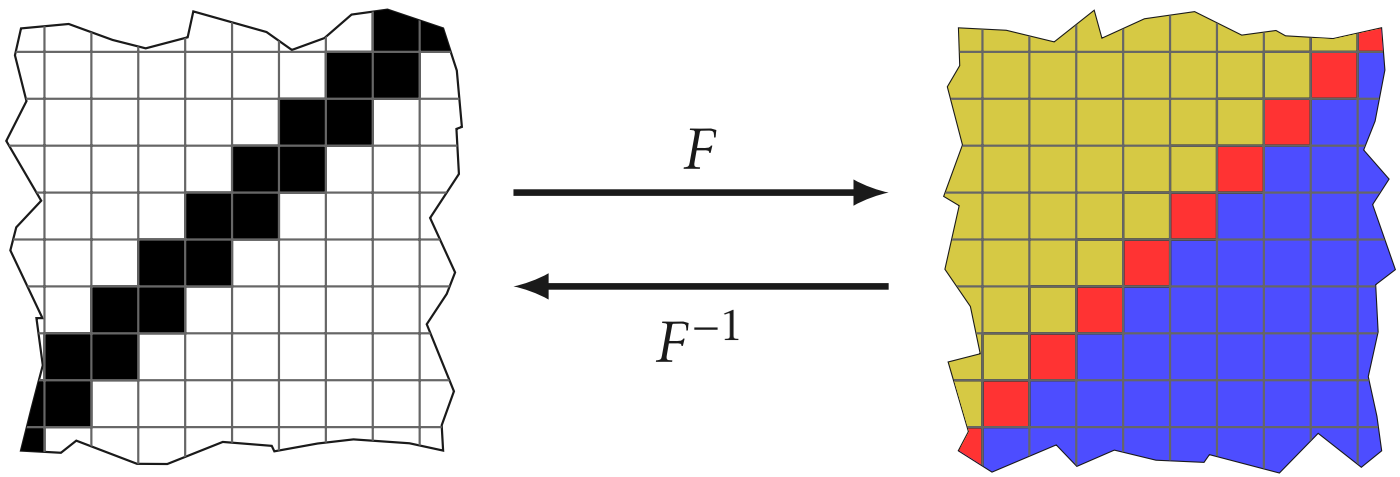
Une fonction de **conjugaison** est une projection locale **réversible** localement.



Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

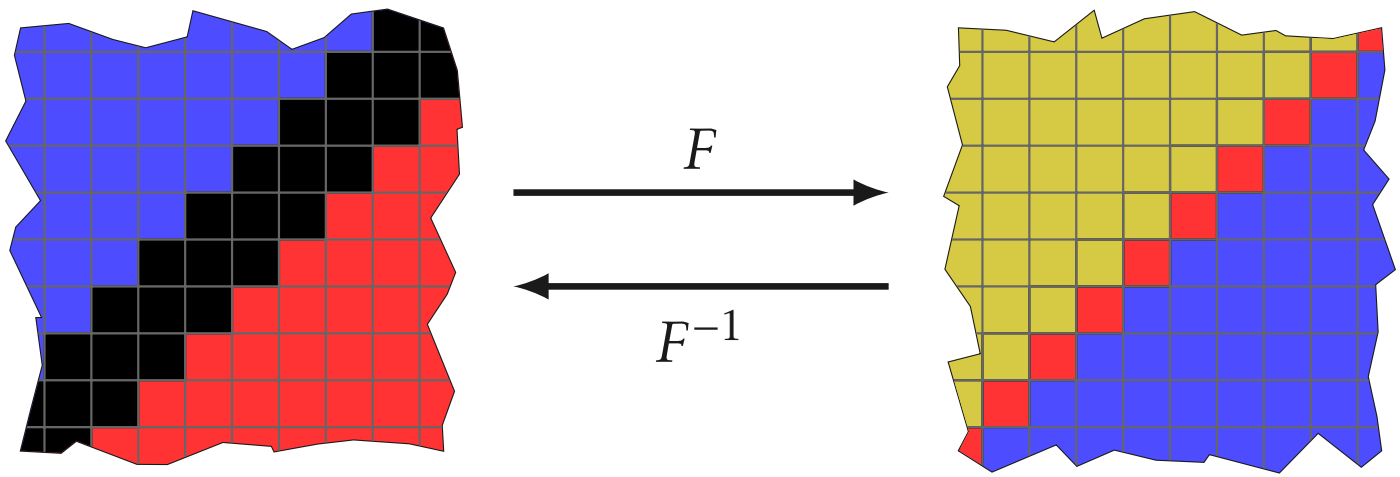
Une fonction de **conjugaison** est une projection locale **réversible** localement.



Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

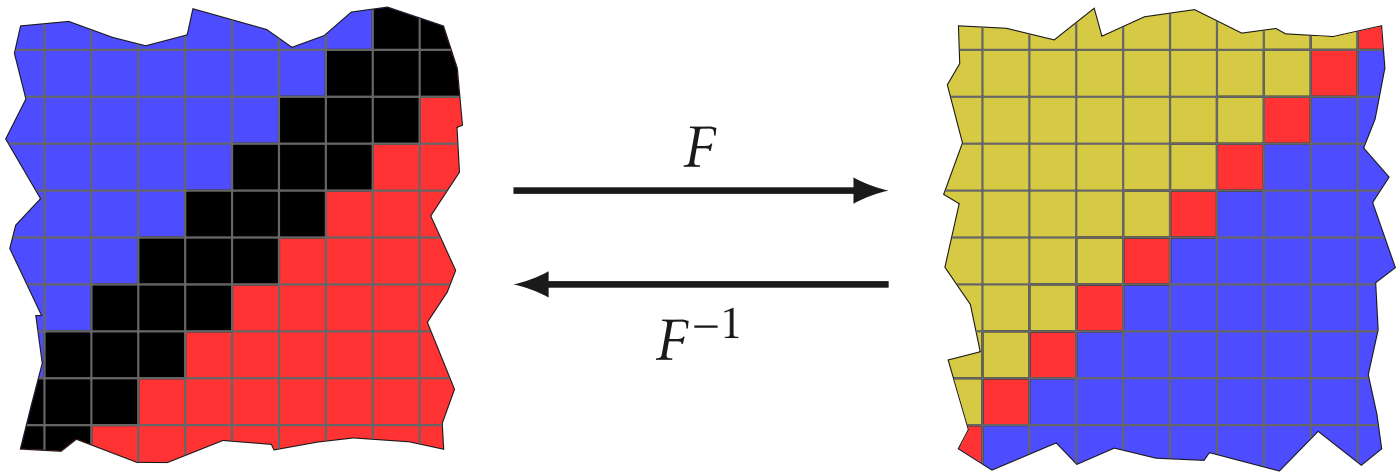
Une fonction de **conjugaison** est une projection locale **réversible** localement.



Conjugaison

Quelle est la **bonne notion d'isomorphisme** entre sous-shifts ?

Une fonction de **conjugaison** est une projection locale **réversible** localement.



Quand la projection n'est **pas réversible**, c'est une fonction de **factorisation**

(In)Décidabilité de la conjugaison

Est-ce qu'on peut décider si deux SFTs sont conjugués ?

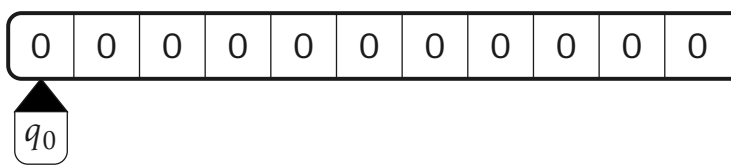
- Le problème est **indécidable en dimension** 2.
- Le problème est **décidable** en dimension 1 **sur** $\mathbb{N}$.
- Le problème est **ouvert** en dimension 1 **sur** $\mathbb{Z}$.

En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_\mathcal{F}$ est **indécidable**.

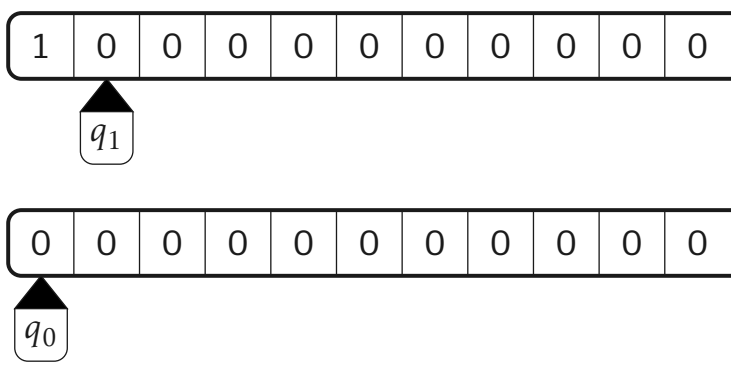
En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_\mathcal{F}$ est **indécidable**.



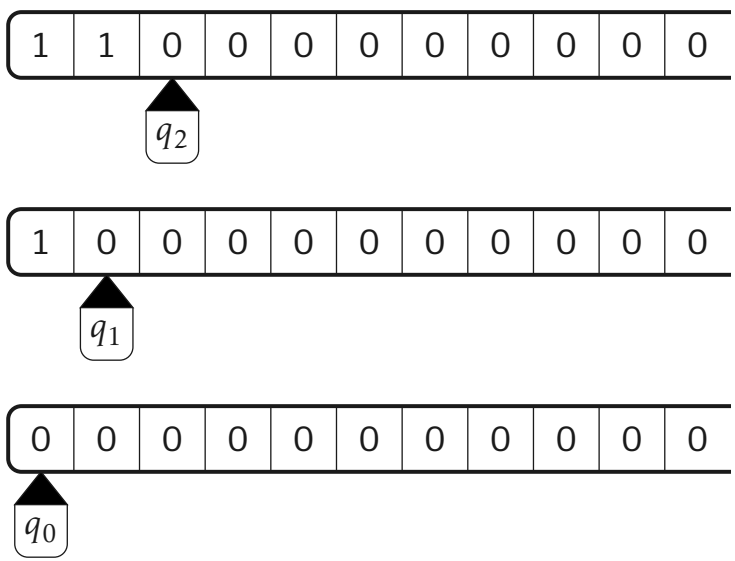
En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_{\mathcal{F}}$ est **indécidable**.



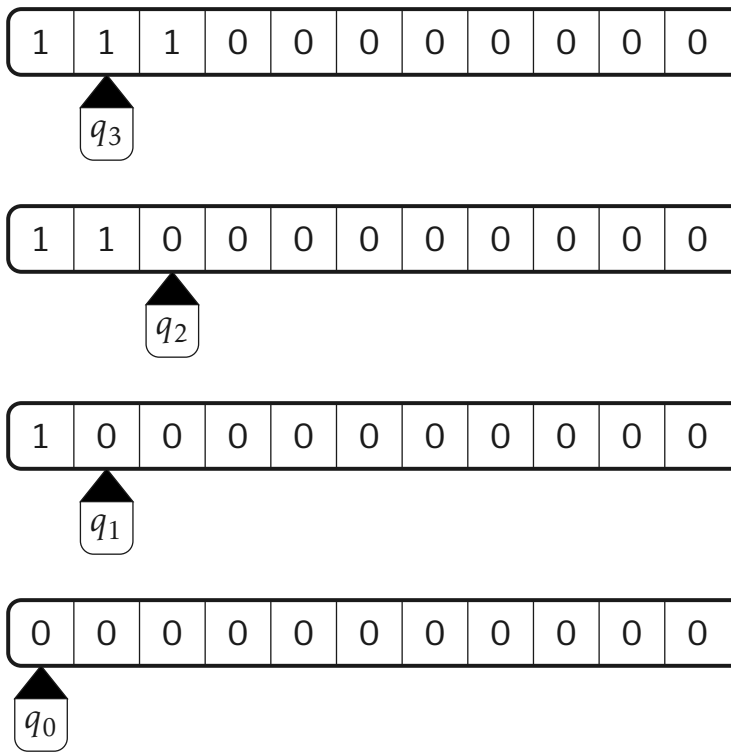
En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_\mathcal{F}$ est **indécidable**.



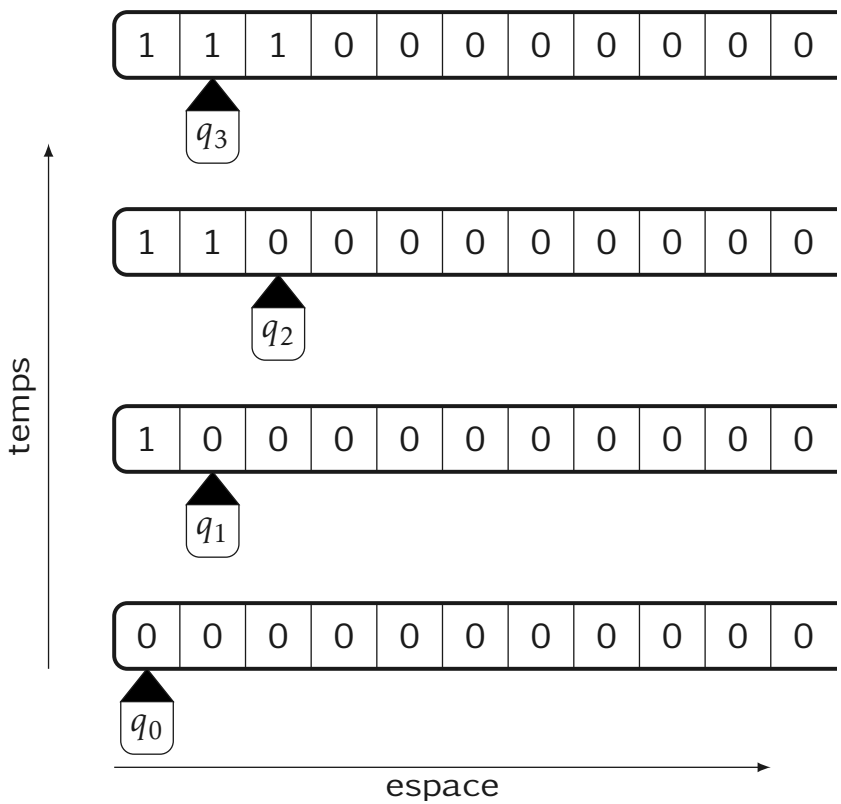
En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_{\mathcal{F}}$ est **indécidable**.



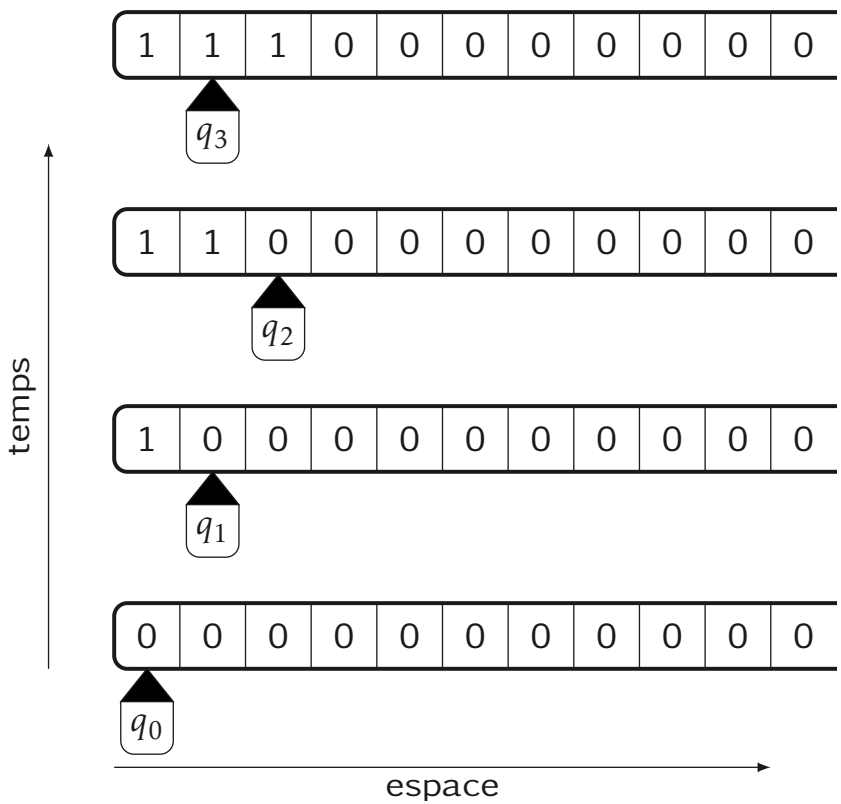
En dimension $d \geq 2$, rien n'est simple

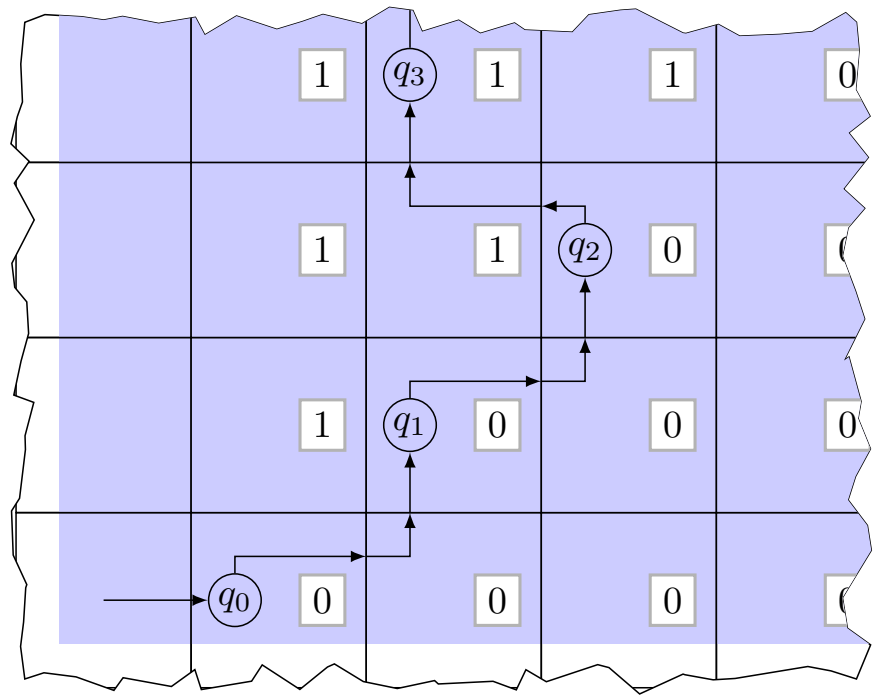
Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_{\mathcal{F}}$ est **indécidable**.



En dimension $d \geq 2$, rien n'est simple

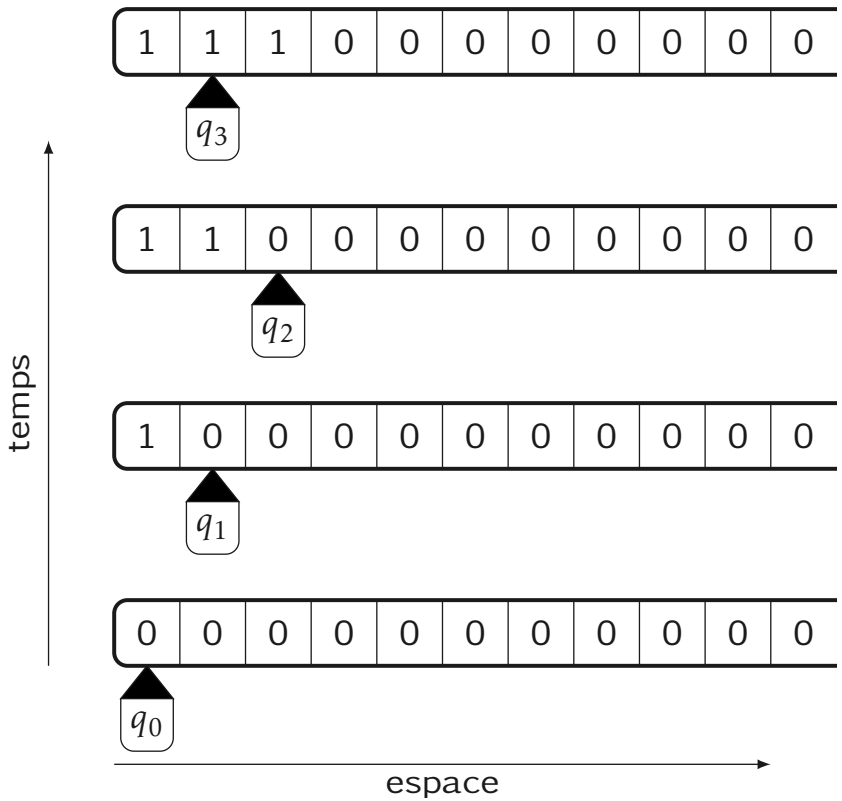
Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_\mathcal{F}$ est **indécidable**.



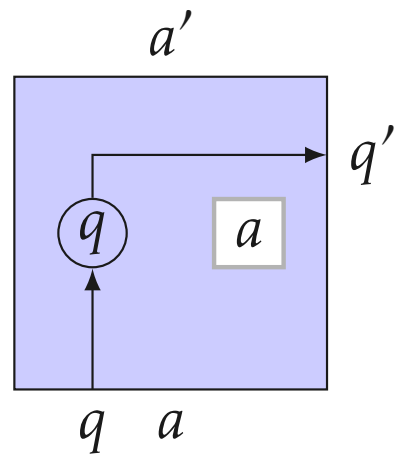


En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_{\mathcal{F}}$ est **indécidable**.

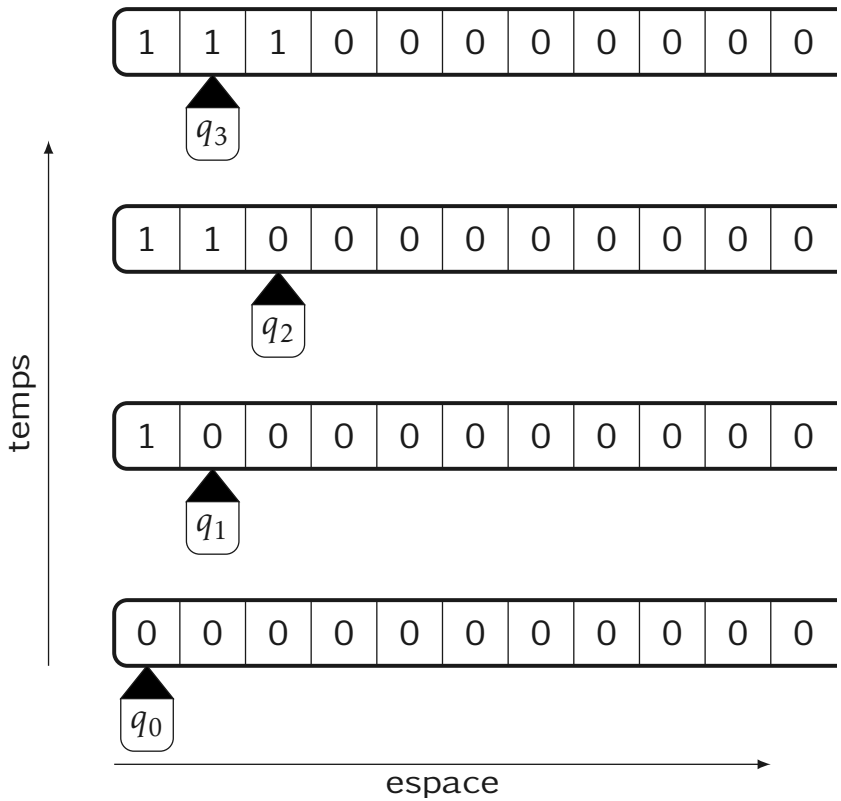


$$(q, a) \longrightarrow (q', a', \rightarrow)$$

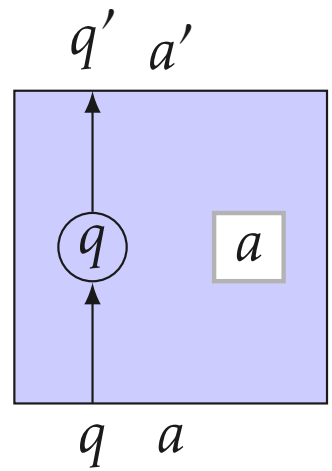


En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_\mathcal{F}$ est **indécidable**.

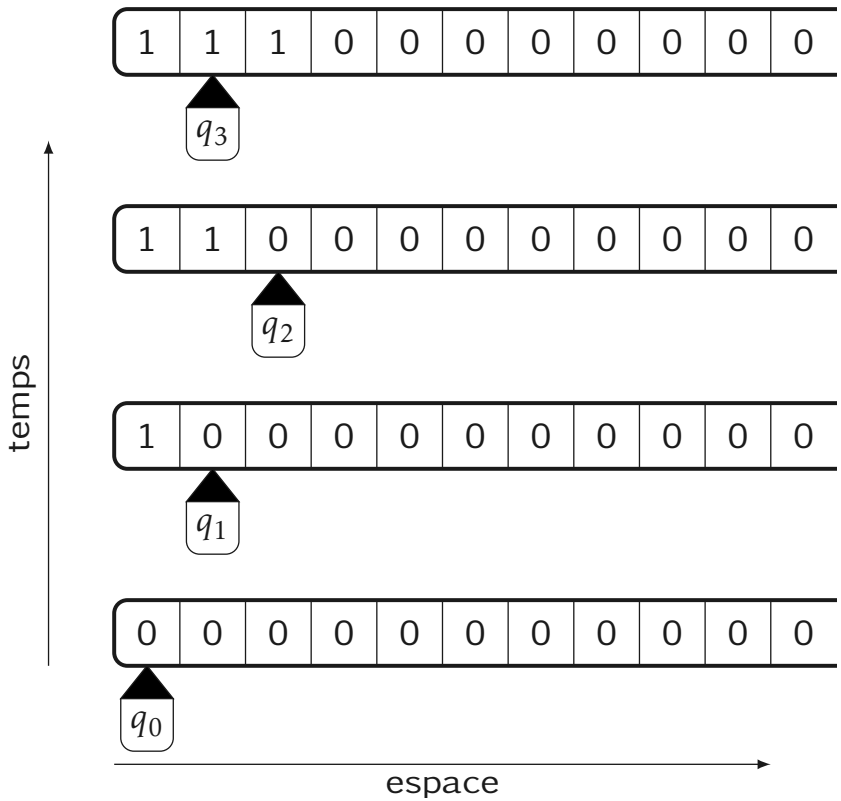


$$(q, a) \longrightarrow (q', a', \uparrow)$$

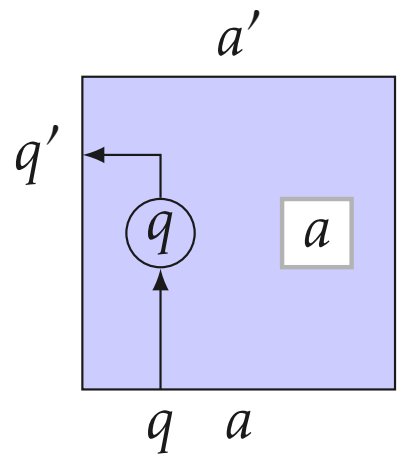


En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_{\mathcal{F}}$ est **indécidable**.

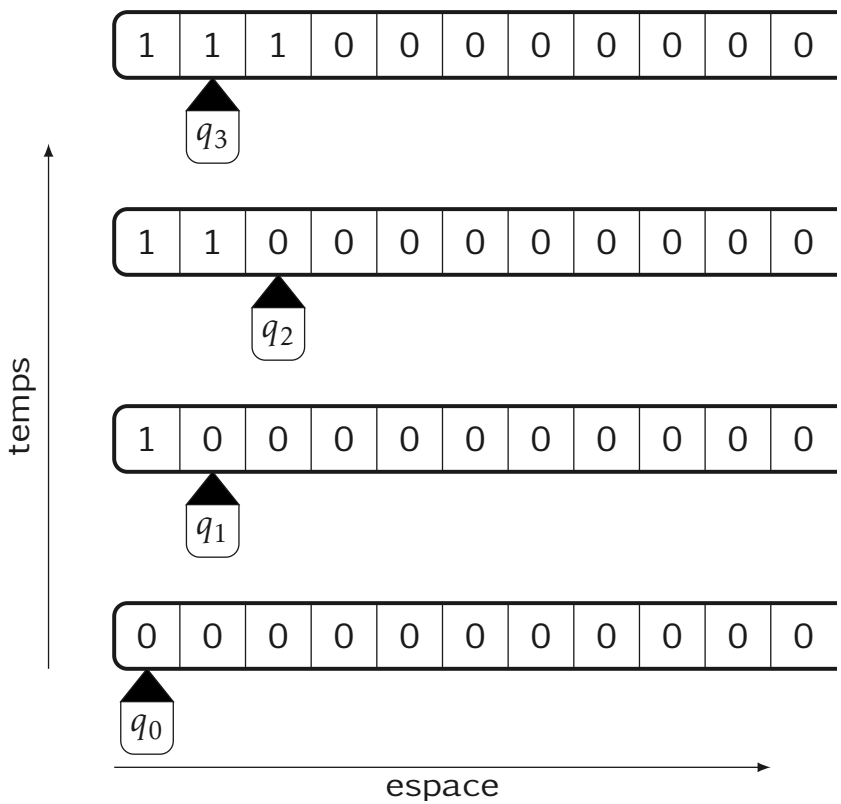


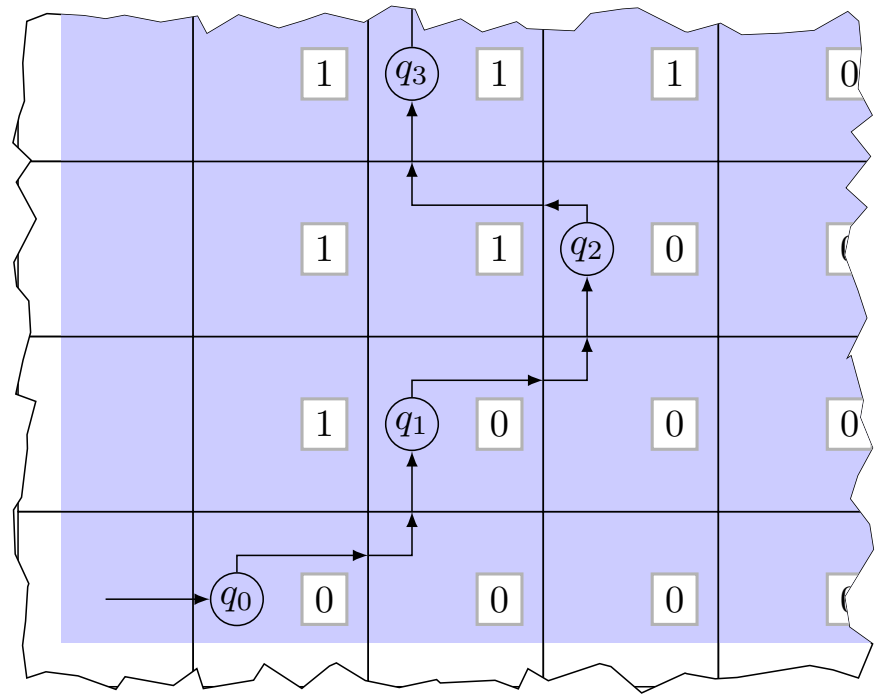
$$(q, a) \longrightarrow (q', a', \leftarrow)$$



En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_\mathcal{F}$ est **indécidable**.

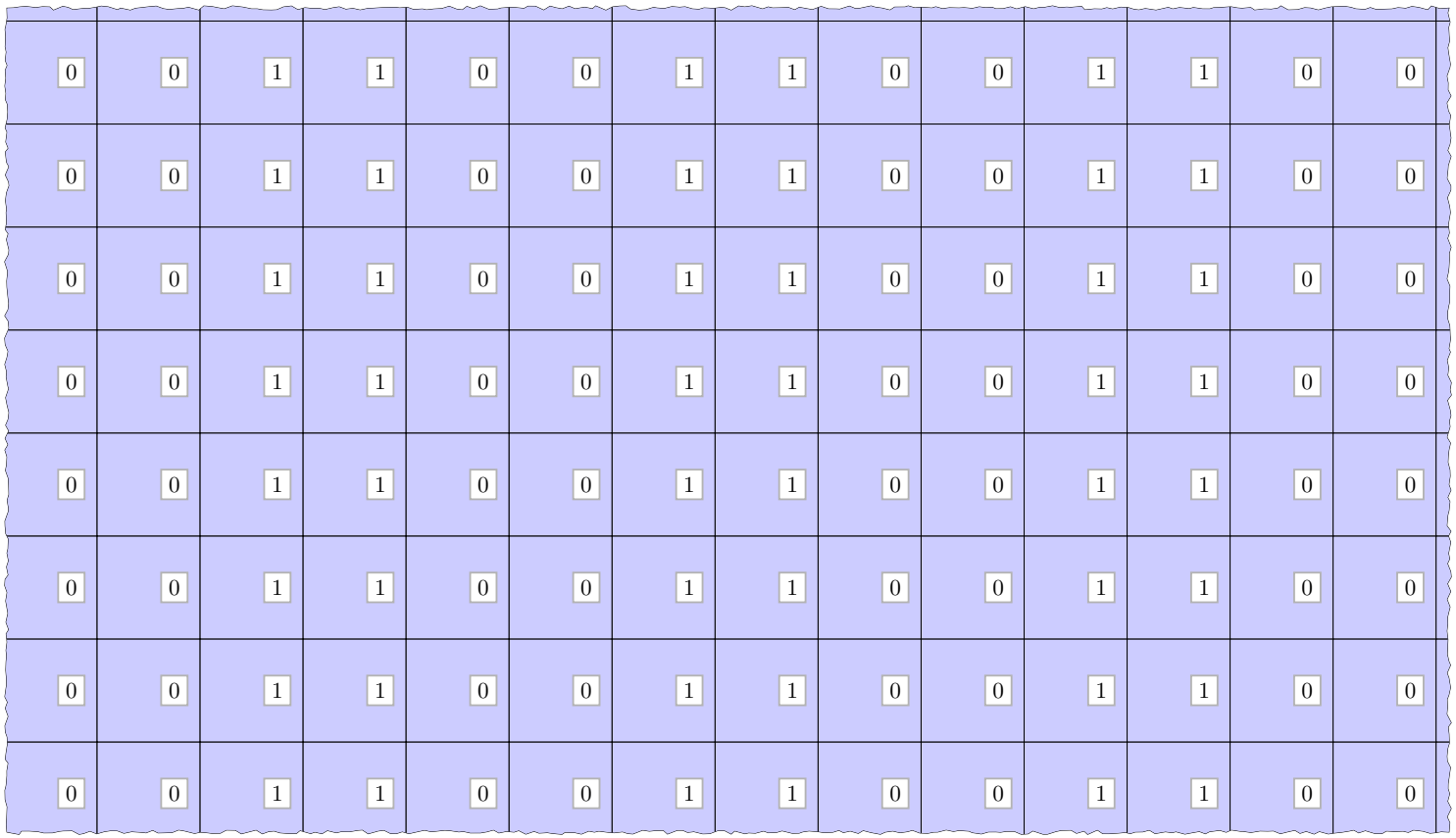




Pavage **infini** $\Leftrightarrow$ la **machine ne s'arrête pas**

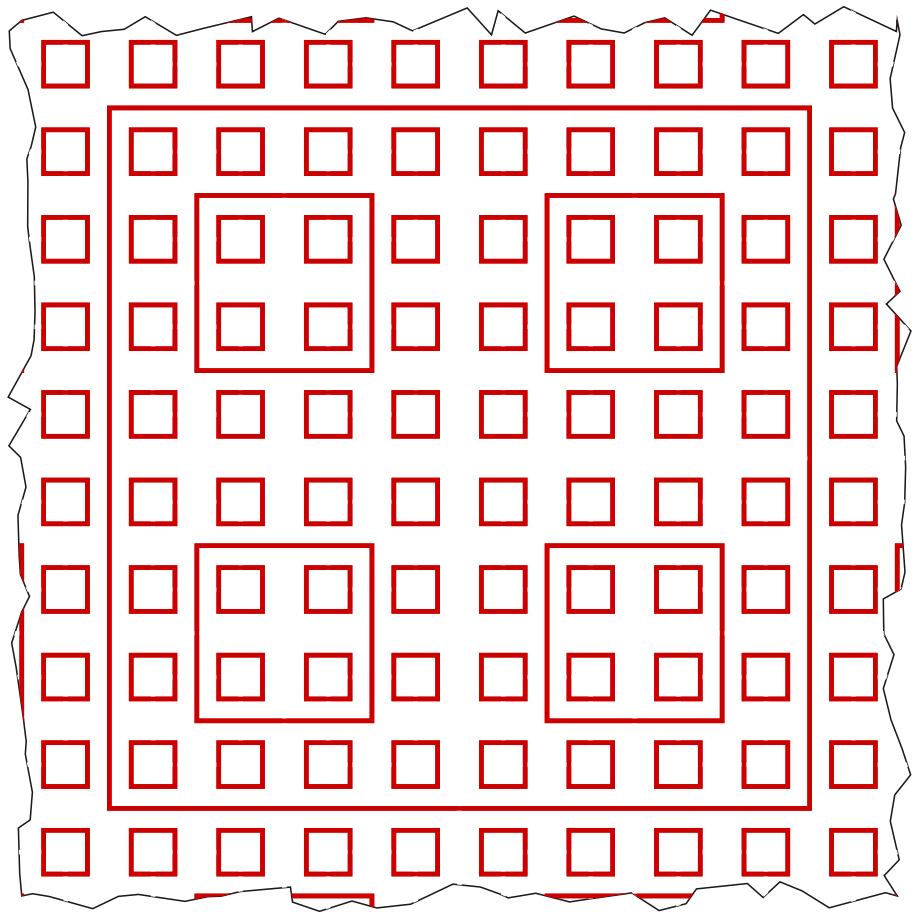
En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_\mathcal{F}$ est **indécidable**.



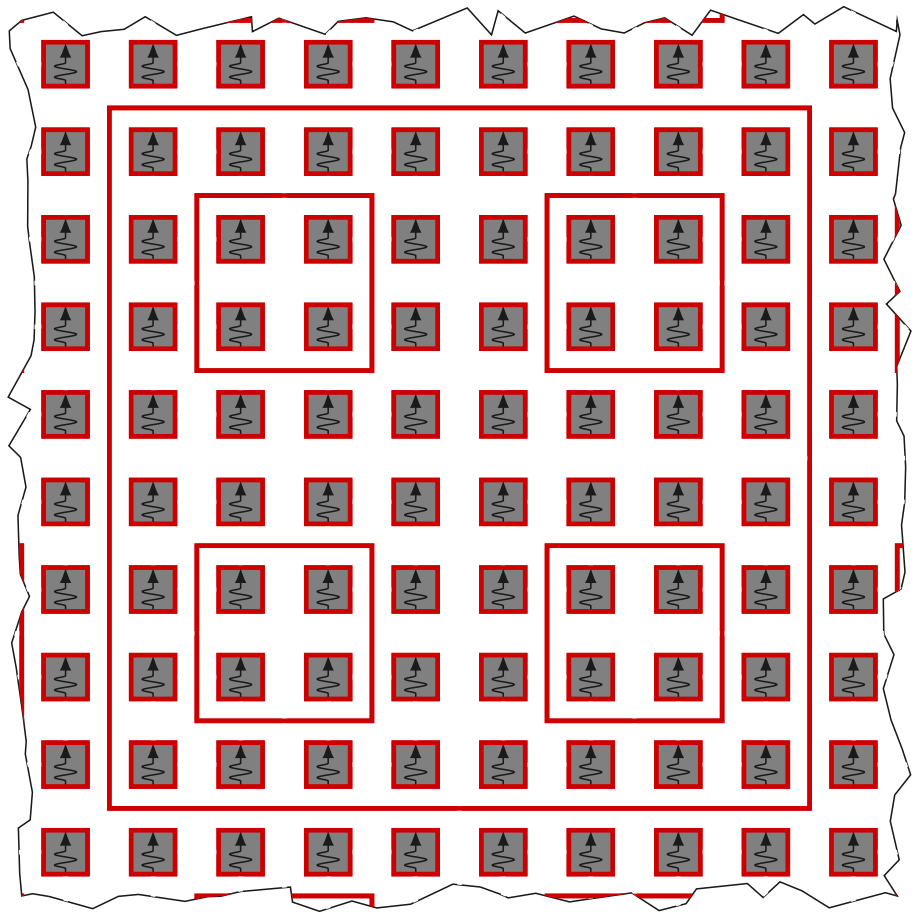
En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_\mathcal{F}$ est **indécidable**.



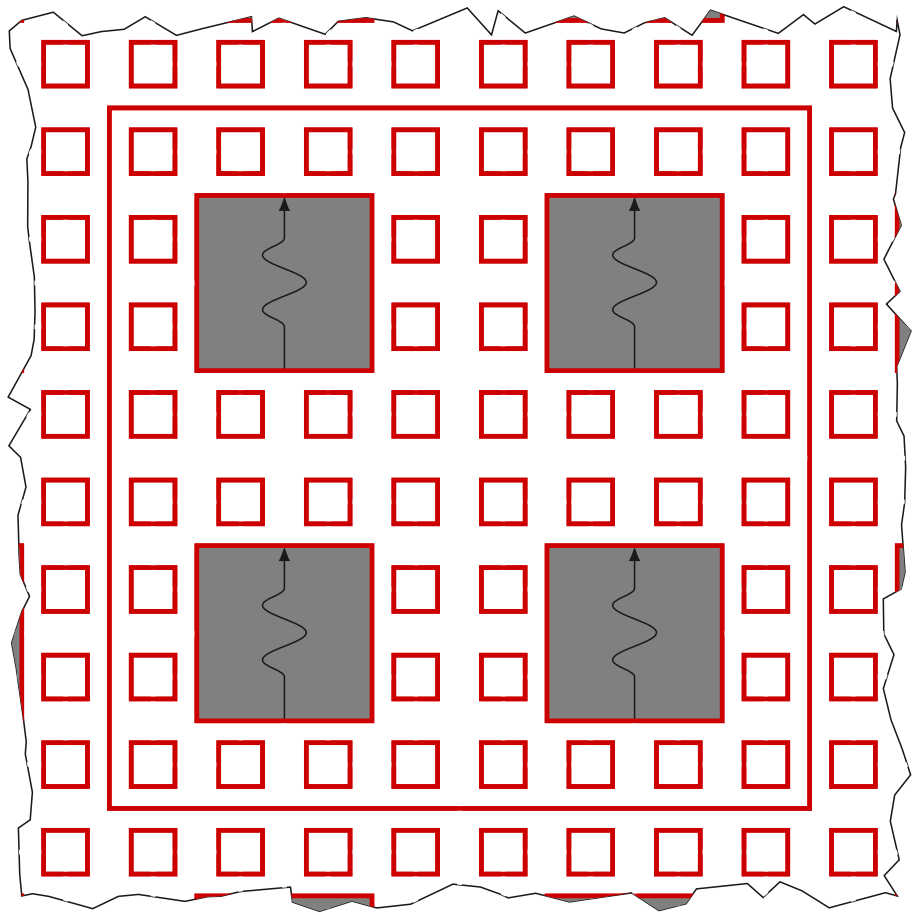
En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_\mathcal{F}$ est **indécidable**.



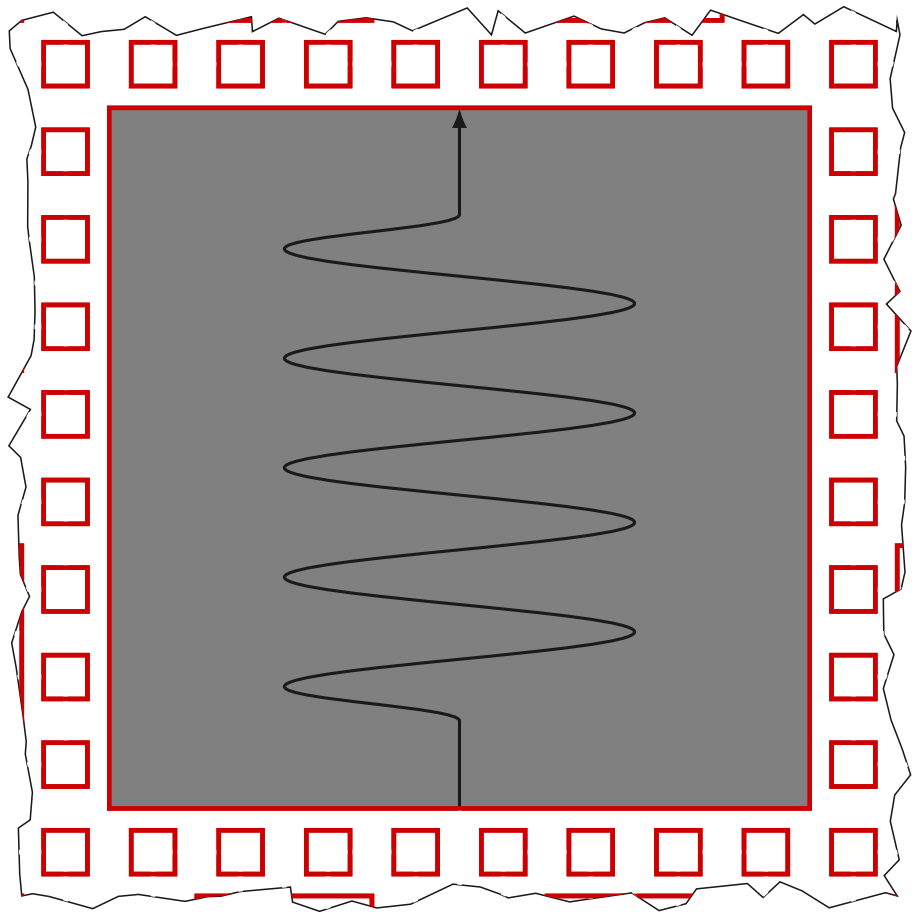
En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_\mathcal{F}$ est **indécidable**.



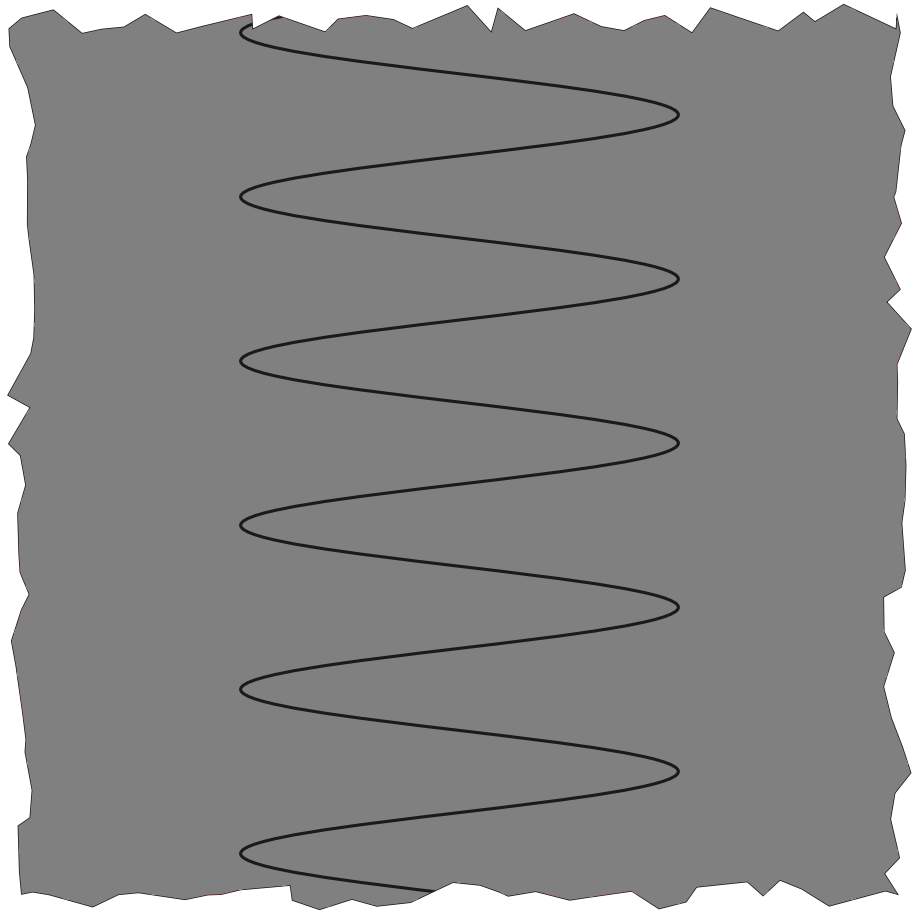
En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_\mathcal{F}$ est **indécidable**.



En dimension $d \geq 2$, rien n'est simple

Théorème [Berger 1964] Étant donné $\mathcal{F}$ en entrée, savoir si il **existe un pavage** dans $\mathcal{X}_\mathcal{F}$ est **indécidable**.



Invariants de conjugaison

Définition Un **invariant de conjugaison** est un objet associé à un SFT qui est le même pour deux SFTs conjugués.

Ils permettent de prouver que deux SFTs ne sont **pas** conjugués.

Invariants classiques :

- Entropie : mesure de la croissance du nombre de motifs [Hochman & Meyerovitch 2010]
- **Nombre de points périodiques**
- Groupe d'automorphismes

Invariants moins classiques :

- Degré Medvedev [Simpson 2012]
- **Degrés Turing**
- Groupe fondamental projectif
- ...

1. Complexité algorithmique de la conjugaison et de la factorisation

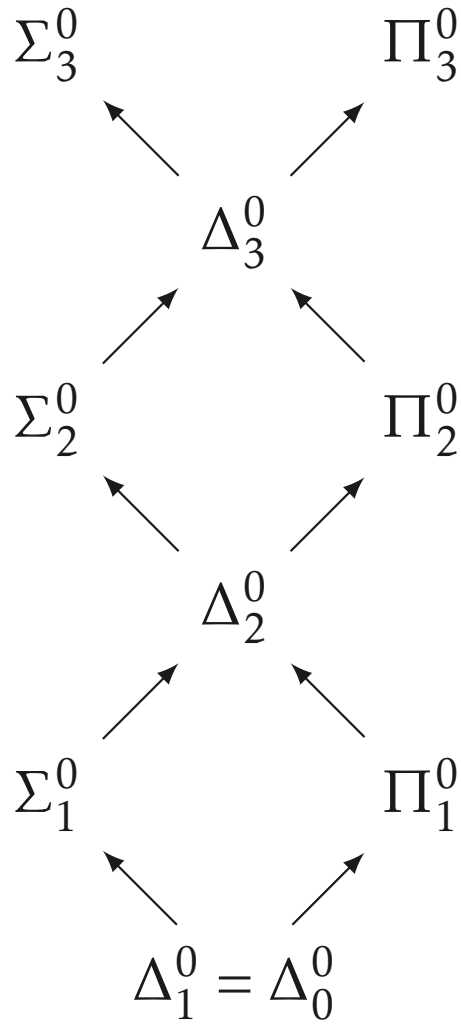
2. Périodicité

3. Degrés Turing des SFTs

Hiérarchie de problèmes indécidables

Définition Un problème $P \subseteq \mathbb{N}$ est Π_n^0 si il existe une machine de Turing M totale telle que

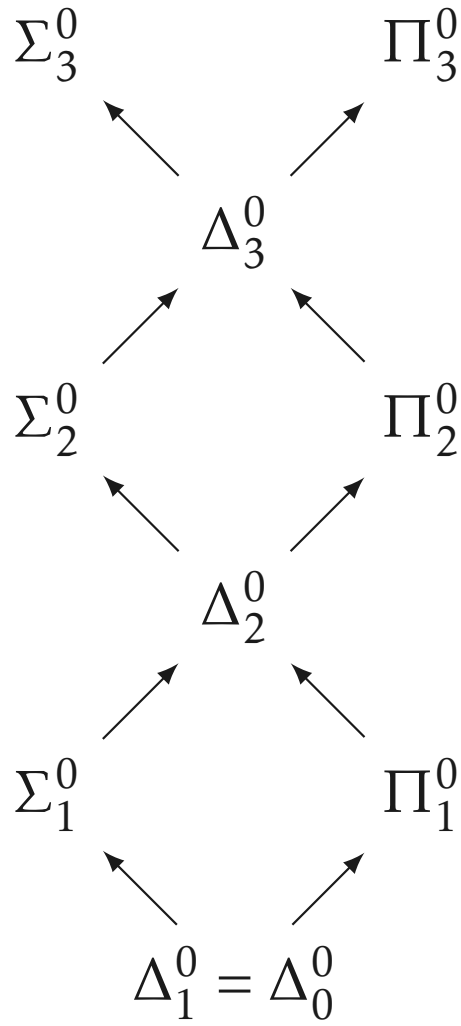
$$n \in P \Leftrightarrow \forall m_1, \exists m_2, \ldots, \Theta m_n, M(n, m_1, \ldots, m_n)$$



Hiérarchie de problèmes indécidables

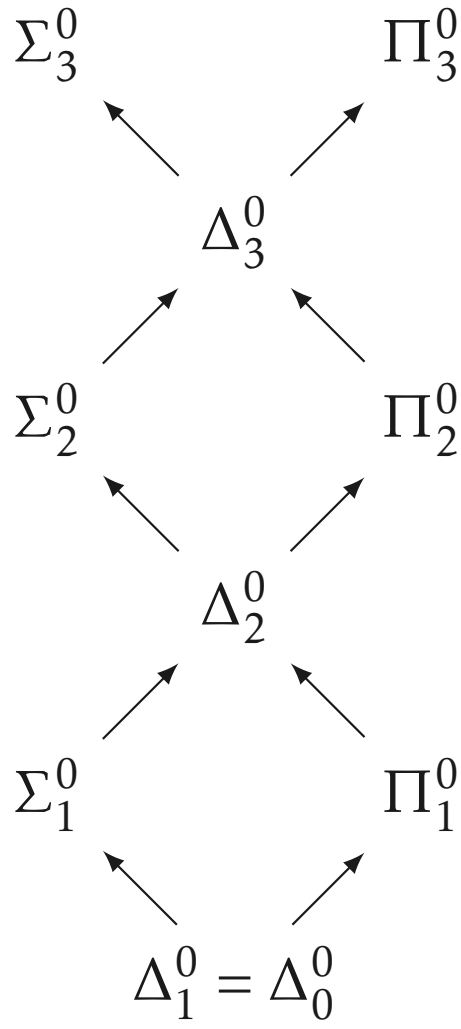
Définition Un problème $P \subseteq \mathbb{N}$ est Σ_n^0 si il existe une machine de Turing M totale telle que

$$n \in P \Leftrightarrow \exists m_1, \forall m_2, \ldots, \Theta m_n, M(n, m_1, \ldots, m_n)$$



Hiérarchie de problèmes indécidables

- Σ_1^0: les ensembles récursivement énumérables
- Π_1^0: les ensembles **co**-récursivement énumérables
- Σ_n^0: les ensembles récursivement énumérables en prenant un ensemble Π_{n-1}^0 en oracle.
- Π_n^0: les ensembles co-récursivement énumérables en Σ_{n-1}^0.

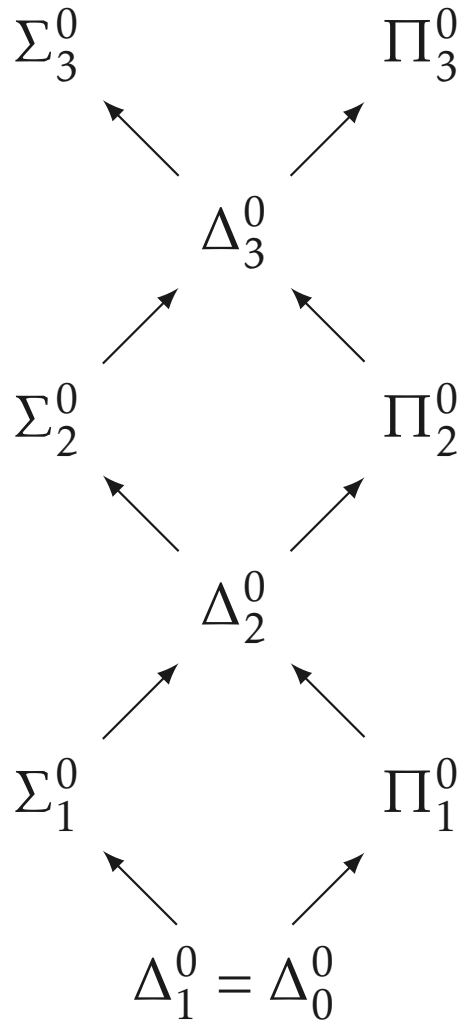


Hiérarchie de problèmes indécidables

Définition Un problème est **complet** pour une classe si il permet de **résoudre tous les problèmes de cette classe**.

Exemples de problèmes complets :

- Σ_1^0 : arrêt d'une machine de Turing (HP)
- Π_2^0 : savoir si une machine de Turing s'arrête sur toutes les entrées (TOT)
- Σ_3^0 : Une machine de Turing ne s'arrête pas uniquement sur un nombre fini d'entrées (COFIN)



Complexité de la conjugaison

Théorème [Jeandel & V.] Étant donné deux **SFTs** X, Y savoir si X est **conjugué** à Y est Σ_1^0**-complet**.

Le théorème est également vrai pour X **fixé**.

Idée de preuve :

Complexité de la conjugaison

Théorème [Jeandel & V.] Étant donné deux **SFTs** X, Y savoir si X est **conjugué** à Y est Σ_1^0**-complet**.

Le théorème est également vrai pour X **fixé**.

Idée de preuve :

La conjugaison est Σ_1^0 :

$$\underbrace{\exists F, G, \underbrace{(F(X) \subseteq Y)}_{\Sigma_1^0} \wedge \underbrace{(G(Y) \subseteq X)}_{\Sigma_1^0} \wedge \underbrace{(F \circ G = \mathrm{id}_X)}_{\Sigma_1^0} \wedge \underbrace{(G \circ F = \mathrm{id}_Y)}_{\Sigma_1^0}}_{\Sigma_1^0}$$

- On devine deux fonctions F et G.
- On vérifie qu'il s'agit bien de fonctions de conjugaison.

Complexité de la conjugaison

Théorème [Jeandel & V.] Étant donné deux **SFTs** X, Y savoir si X est **conjugué** à Y est Σ_1^0**-complet**.

Le théorème est également vrai pour X **fixé**.

Idée de preuve :

La conjugaison est Σ_1^0-dure, réduction à partir de SFT vide :

- R_M un SFT vide ssi M s'arrête.
- n plus grand que le nombre de couleurs de X.

$$X \overset{?}{\cong} X \sqcup R_M \times \{0,...,n\}^{\mathbb{Z}^2}$$

- Si R_M est vide, alors X et $X \sqcup R_M \times \{0,\dots,n\}^{\mathbb{Z}^2}$ sont égaux.
- Sinon X et $X \sqcup R_M \times \{0,\dots,n\}^{\mathbb{Z}^2}$ ne sont pas conjugués.

□

Complexité de la conjugaison (Sofiques)

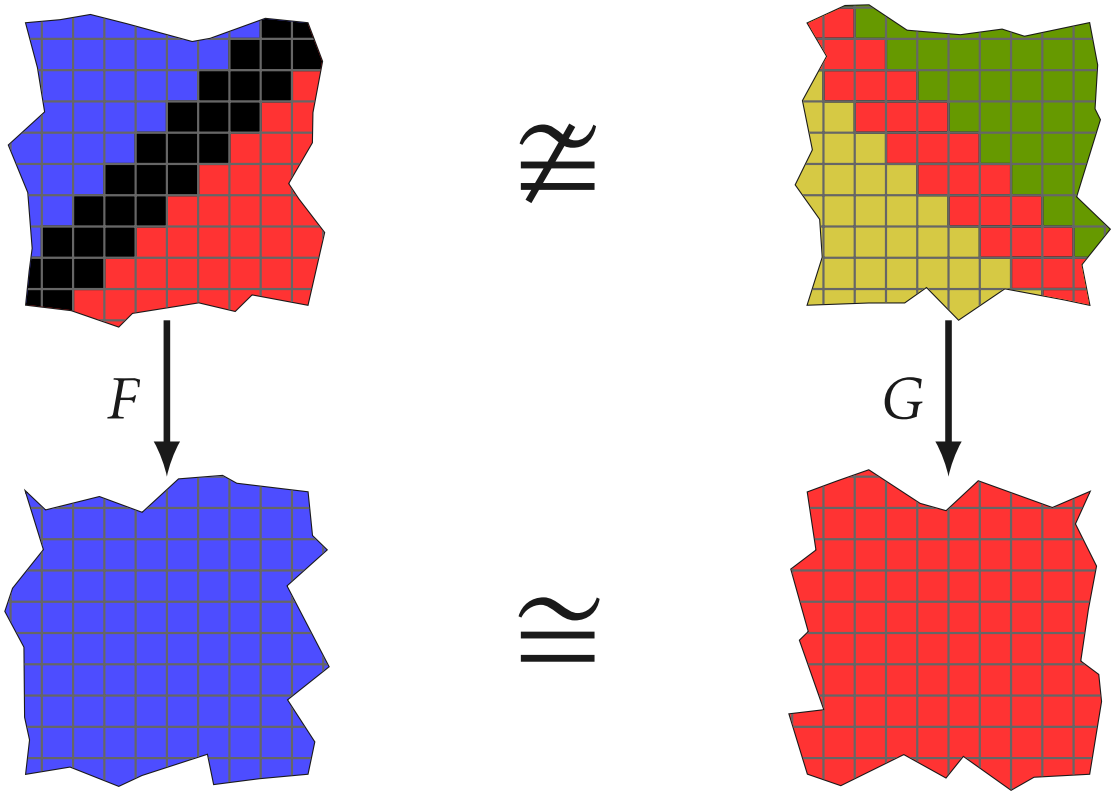
Théorème [Jeandel & V.] Étant donnés deux sous-shifts **sofiques** en entrée, savoir si ils sont **conjugués** est Σ_3^0**-complet**.

Savoir si $F(X) \subseteq Y$ devient Π_2^0.

Complexité de la conjugaison (Sofiques)

Théorème [Jeandel & V.] Étant donnés deux sous-shifts **sofiques** en entrée, savoir si ils sont **conjugués** est Σ_3^0**-complet**.

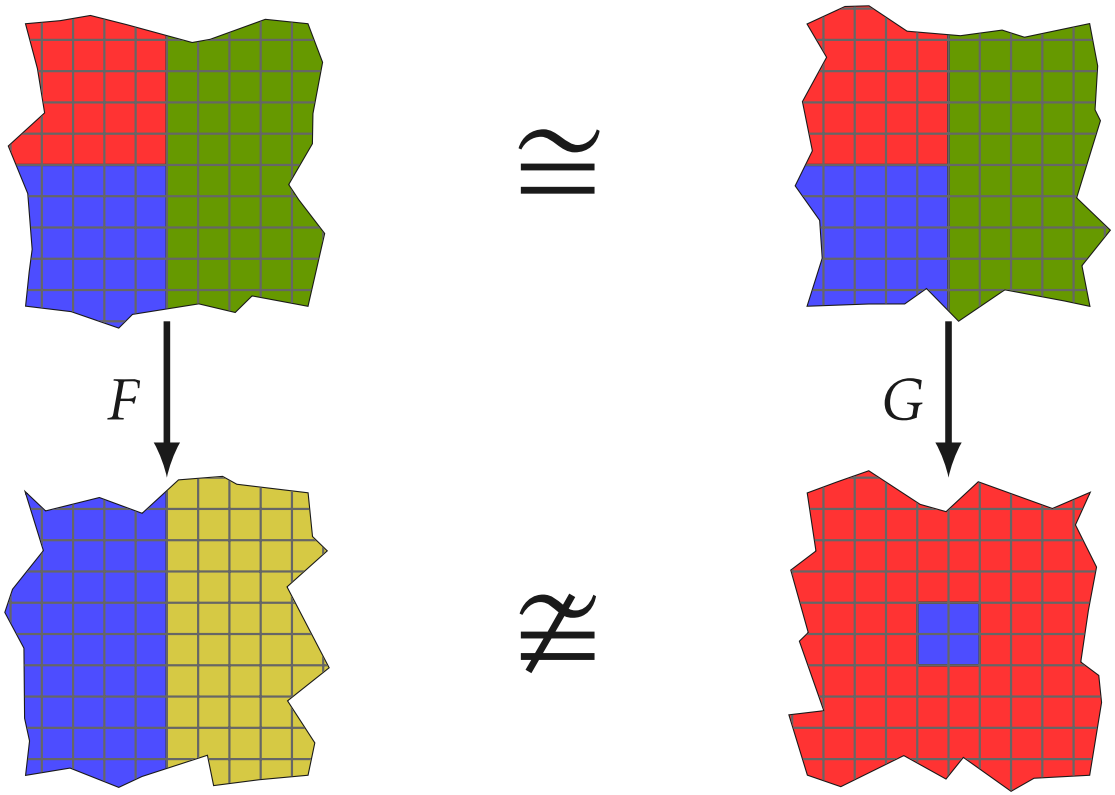
Savoir si $F(X) \subseteq Y$ devient Π_2^0.



Complexité de la conjugaison (Sofiques)

Théorème [Jeandel & V.] Étant donnés deux sous-shifts **sofiques** en entrée, savoir si ils sont **conjugués** est Σ_3^0**-complet**.

Savoir si $F(X) \subseteq Y$ devient Π_2^0.



Complexité de la factorisation

Théorème [Jeandel & V.] Étant donnés deux **SFTs** X, Y, savoir si X se **factorise** sur Y est Σ_3^0**-complet**.

Théorème [Jeandel & V.] Étant donnés deux sous-shifts **sofiques** X, Y, savoir si X se **factorise** sur Y est Σ_3^0**-complet**.

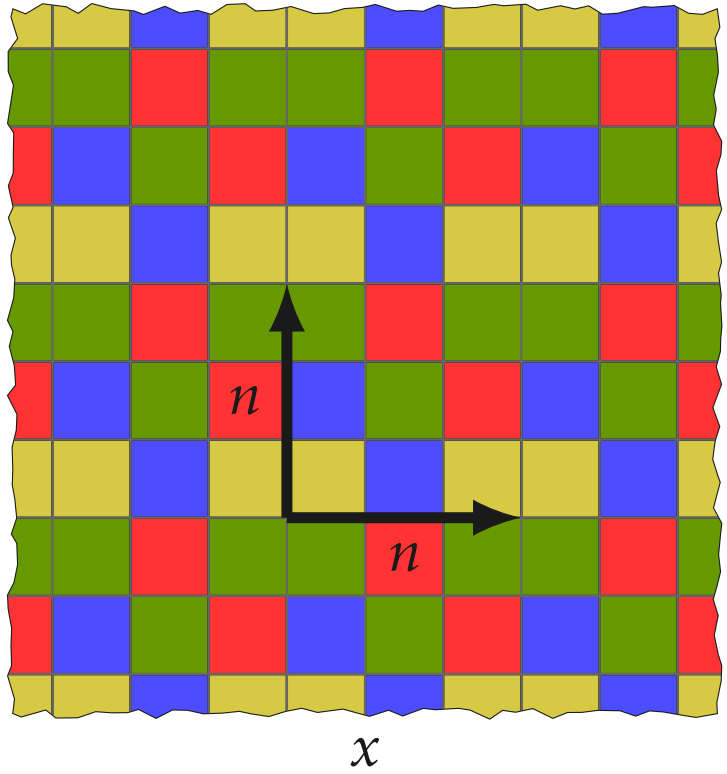
1. Complexité algorithmique de la conjugaison et de la factorisation

2. Périodicité

3. Degrés Turing des SFTs

Périodicité

x est **périodique** de **période** n si $\sigma_{n\boldsymbol{e_i}}(x) = x$ pour tout $1 \leq i \leq d$.



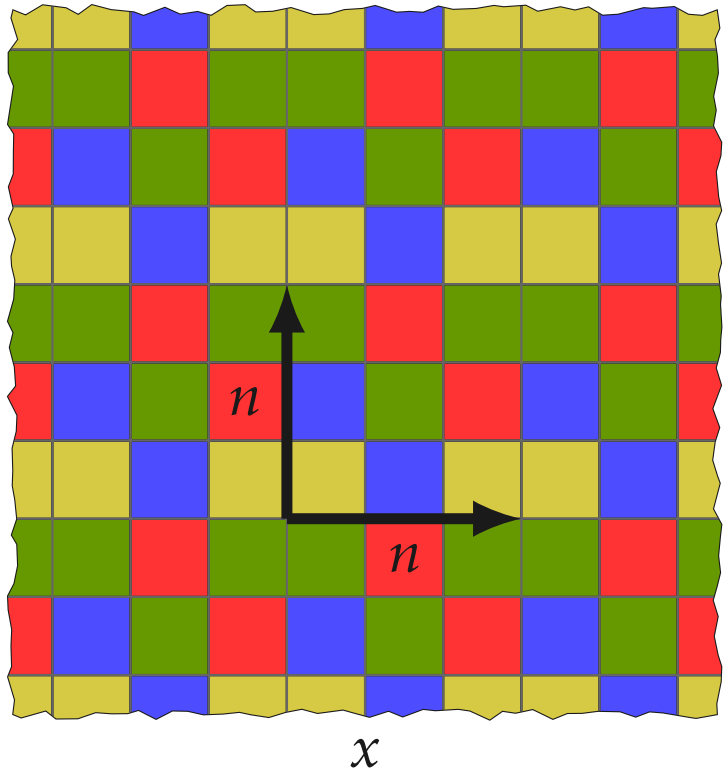
Un seul motif carré qui est répété partout.

Descriptible par une information **finie**.

Période : le **plus petit** $n \in \mathbb{N}^*$.

Périodicité

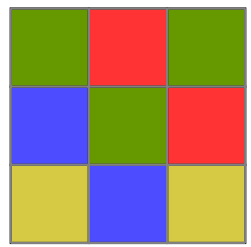
x est **périodique** de **période** n si $\sigma_{ne_i}(x) = x$ pour tout $1 \leq i \leq d$.



Un seul motif carré qui est répété partout.

Descriptible par une information **finie**.

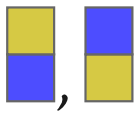
Période : le **plus petit** $n \in \mathbb{N}^*$.

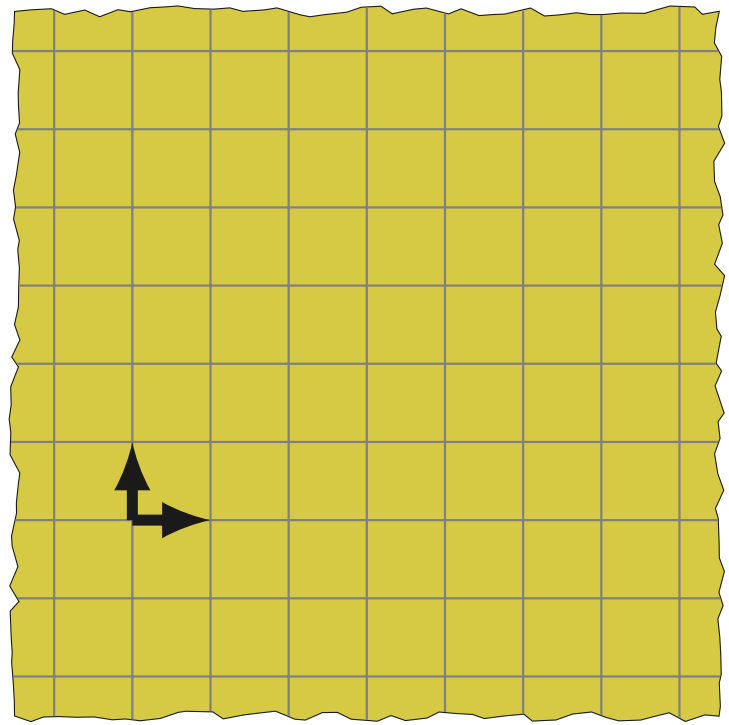


Ensembles de périodes

On note $\mathfrak{P}_X$ l'ensemble des périodes de X.

$\Sigma = \{$  $\}$

$\mathcal{F} = \left\{$  $\right\}$



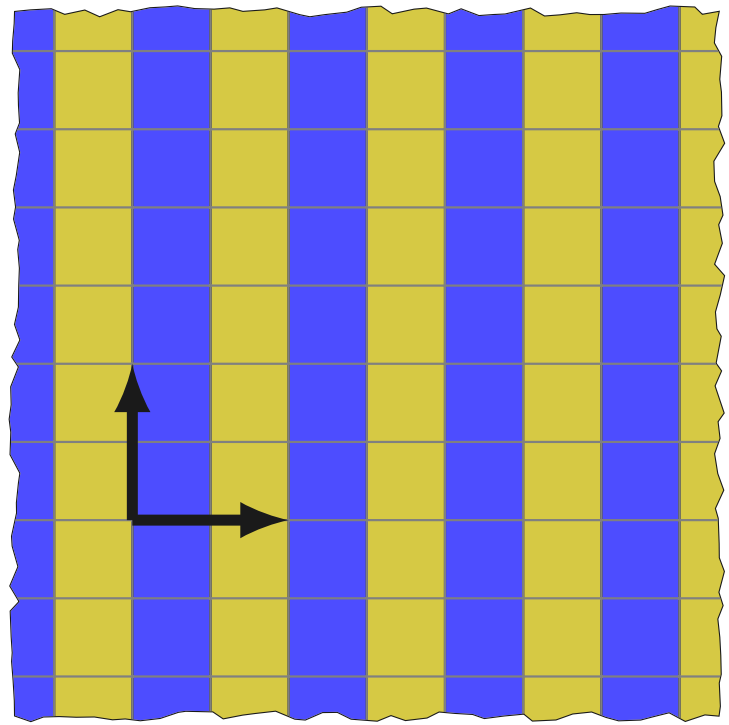
$1 \in \mathfrak{P}_{\mathcal{X}_\mathcal{F}}$

Ensembles de périodes

On note $\mathfrak{P}_X$ l'ensemble des périodes de X.

$\Sigma = \{$  $\}$

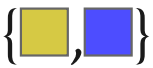
$\mathcal{F} = \left\{$  $\right\}$

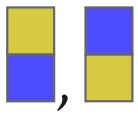


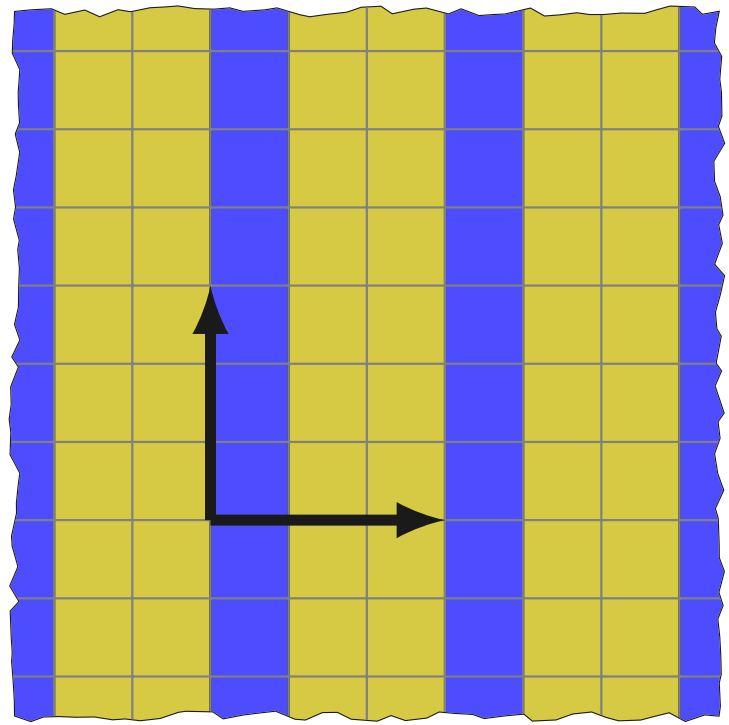
$2 \in \mathfrak{P}_{\mathcal{X}_\mathcal{F}}$

Ensembles de périodes

On note $\mathfrak{P}_X$ l'ensemble des périodes de X.

$\Sigma = \{$  $\}$

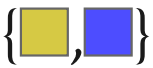
$\mathcal{F} = \left\{$  $\right\}$

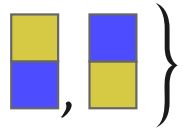


$3 \in \mathfrak{P}_{\mathcal{X}_\mathcal{F}}$

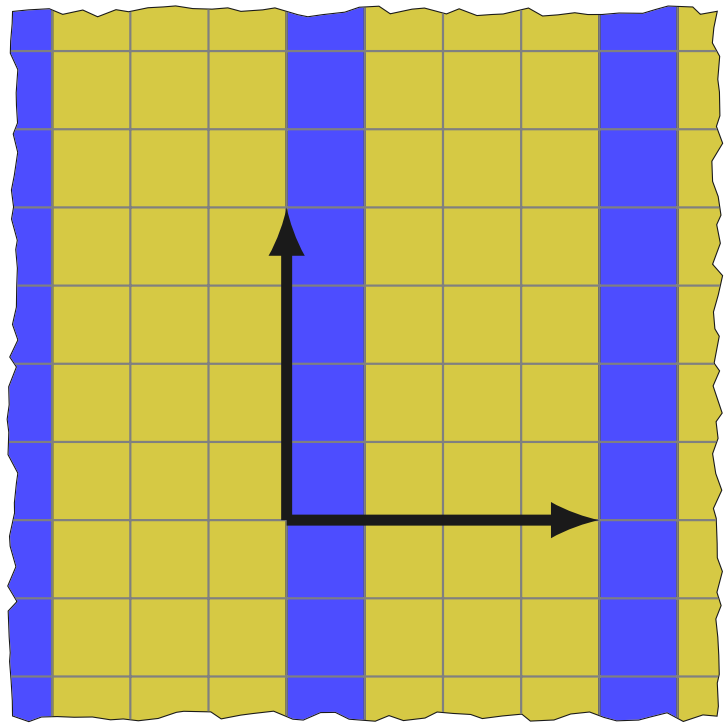
Ensembles de périodes

On note $\mathfrak{P}_X$ l'ensemble des périodes de X.

$\Sigma = \{$  $\}$

$\mathcal{F} =$ 

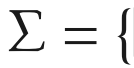
$\mathfrak{P}_{\mathcal{X}_\mathcal{F}} = \mathbb{N}^*$

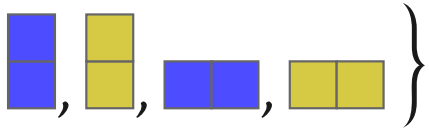


$4 \in \mathfrak{P}_{\mathcal{X}_\mathcal{F}}$

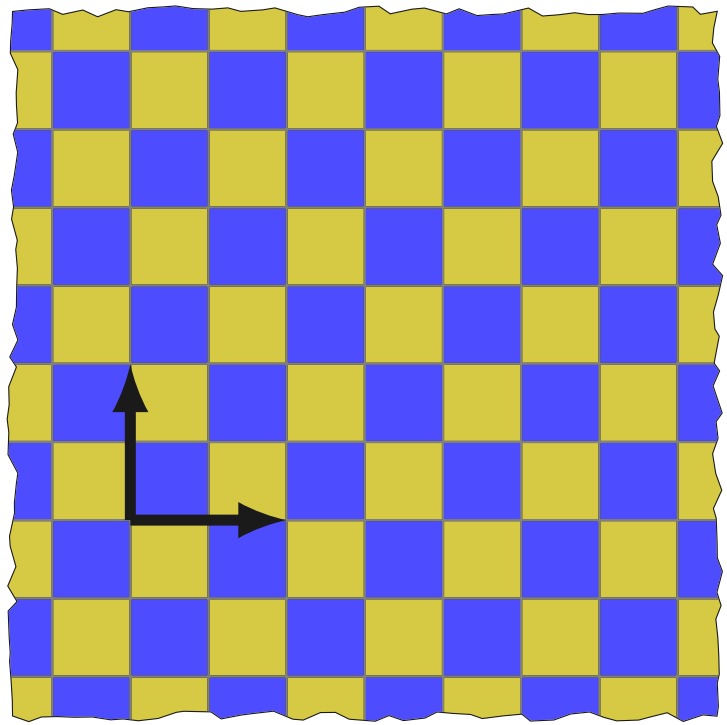
Ensembles de périodes

On note $\mathfrak{P}_X$ l'ensemble des périodes de X.

$\Sigma =$ 

$\mathcal{F} =$ 

$\mathfrak{P}_{\mathcal{X}_\mathcal{F}} = \{2\}$



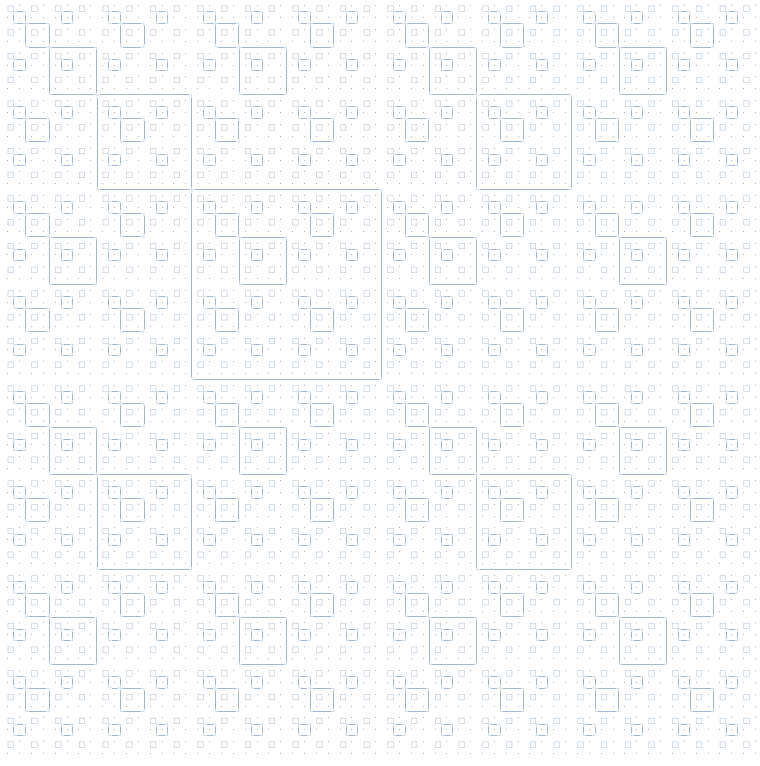
Uniquement des damiers

Ensembles de périodes

On note $\mathfrak{P}_X$ l'ensemble des périodes de X.

[Berger 1964] Il existe un SFT $\mathcal{A}$ ne contenant **aucun point périodique**.

$$\mathfrak{P}_{\mathcal{A}} = \emptyset$$



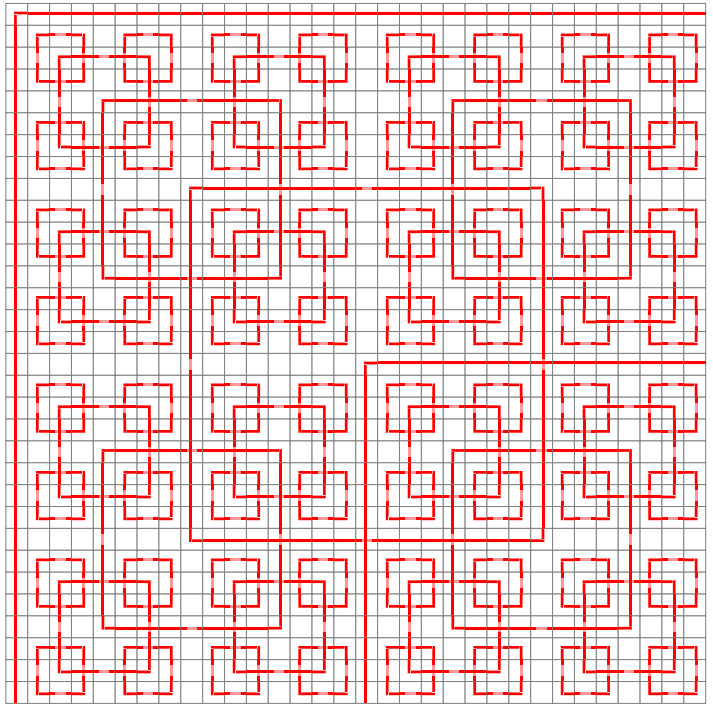
[Berger 1964]

Ensembles de périodes

On note $\mathfrak{P}_X$ l'ensemble des périodes de X.

[Berger 1964] Il existe un SFT $\mathcal{A}$ ne contenant **aucun point périodique**.

$$\mathfrak{P}_{\mathcal{A}} = \emptyset$$



[Robinson 1971]

Ensembles de périodes

On note $\mathfrak{P}_X$ l'ensemble des périodes de X.

[Berger 1964] Il existe un SFT $\mathcal{A}$ ne contenant **aucun point périodique**.

$$\mathfrak{P}_{\mathcal{A}} = \emptyset$$

Et de nombreux autres :

[Knuth 1968]

[Anderaa & Lewis 1974]

[Kari 1996]

[Ollinger 2008]

[Poupet 2010]

...

Ensembles de périodes

On note $\mathfrak{P}_X$ l'ensemble des périodes de X.

L'ensemble des périodes est un **invariant de conjugaison**.

Que peut-on dire de $\mathfrak{P}_X$?

Théorème [Gurevich & Koryakov 1972] Savoir si $\mathfrak{P}_X$ est vide indécidable.

Que peut-on dire de $\mathfrak{P}_X$?

Théorème [Gurevich & Koryakov 1972] Savoir si $\mathfrak{P}_X$ est vide **indécidable**.

Théorème Étant donné n, on peut décider si il existe une configuration de période n

Preuve : En dimension 2, deviner un motif $n \times n$ et vérifier les contraintes.

Que peut-on dire de $\mathfrak{P}_X$?

Théorème [Gurevich & Koryakov 1972] Savoir si $\mathfrak{P}_X$ est vide **indécidable**.

Théorème Étant donné n, on peut décider si il existe une configuration de période n

Preuve : En dimension 2, deviner un motif $n \times n$ et vérifier les contraintes. Se fait en temps n^2 non-déterministe.

Que peut-on dire de $\mathfrak{P}_X$?

Théorème [Gurevich & Koryakov 1972] Savoir si $\mathfrak{P}_X$ est vide **indécidable**.

Théorème Étant donné n, on peut décider si il existe une configuration de période n en **temps NE**, quand n **est codé en binaire**.

Preuve : En dimension 2, deviner un motif $n \times n$ et vérifier les contraintes. Se fait en temps n^2 non-déterministe.

$\mathbf{NE} = \bigcup_{k \in \mathbb{N}} \mathbf{NTIME}(2^{k|n|})$: Langages reconnus en temps exponentiel linéaire. $\mathbf{NE} = \mathbf{NP}_1 = \bigcup_{k \in \mathbb{N}} \mathbf{NTIME}(n^k)$, la classe $\mathbf{NP}$ sur langages codés en unaire.

Quels ensembles peut-on réaliser ?

Théorème [Jeandel & V.] Pour tout langage $L \in \mathbf{NE}$, on peut construire un SFT X dont les périodes sont exactement L.

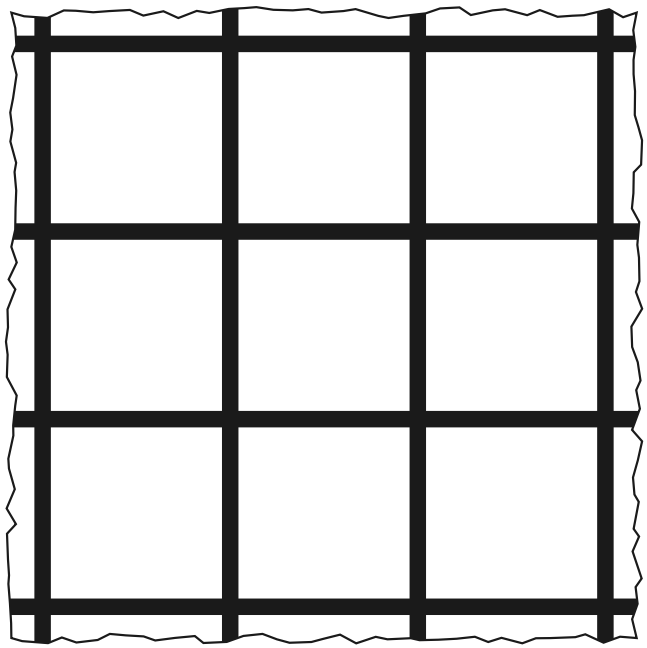
Remarque X n'est pas nécessairement de dimension 2.

Quels ensembles peut-on réaliser ?

Théorème [Jeandel & V.] Pour tout langage $L \in \mathbf{NE}$, on peut construire un SFT X dont les périodes sont exactement L.

Remarque X n'est pas nécessairement de dimension 2.

Idée de la preuve. Soit L un langage de $\mathbf{NTIME}_1(n)$:

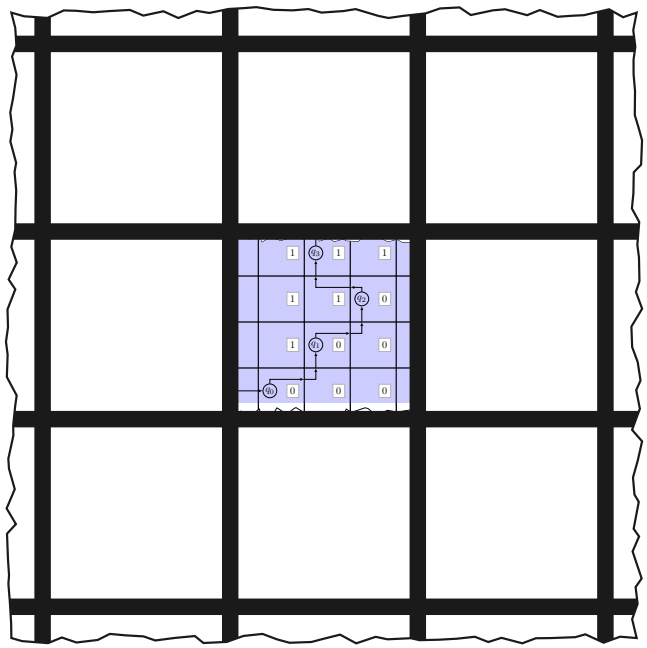


Quels ensembles peut-on réaliser ?

Théorème [Jeandel & V.] Pour tout langage $L \in \mathbf{NE}$, on peut construire un SFT X dont les périodes sont exactement L.

Remarque X n'est pas nécessairement de dimension 2.

Idée de la preuve. Soit L un langage de $\mathbf{NTIME}_1(n)$:

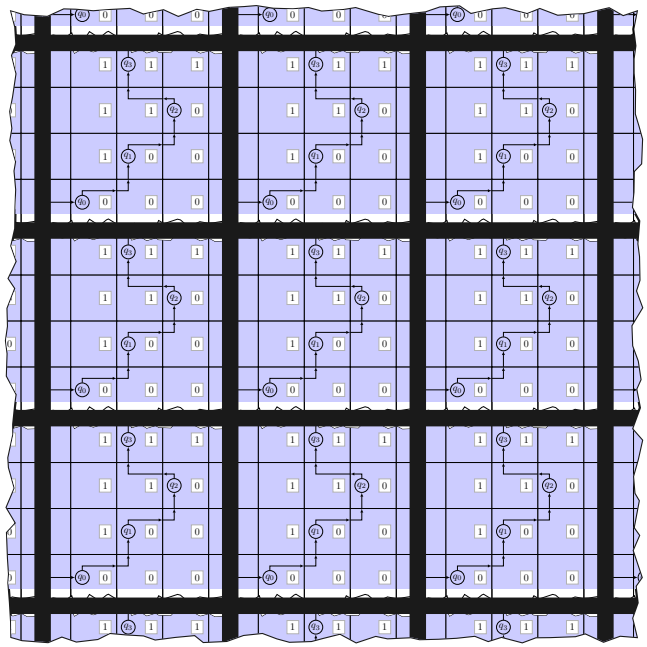


Quels ensembles peut-on réaliser ?

Théorème [Jeandel & V.] Pour tout langage $L \in \mathbf{NE}$, on peut construire un SFT X dont les périodes sont exactement L.

Remarque X n'est pas nécessairement de dimension 2.

Idée de la preuve. Soit L un langage de $\mathbf{NTIME}_1(n)$:

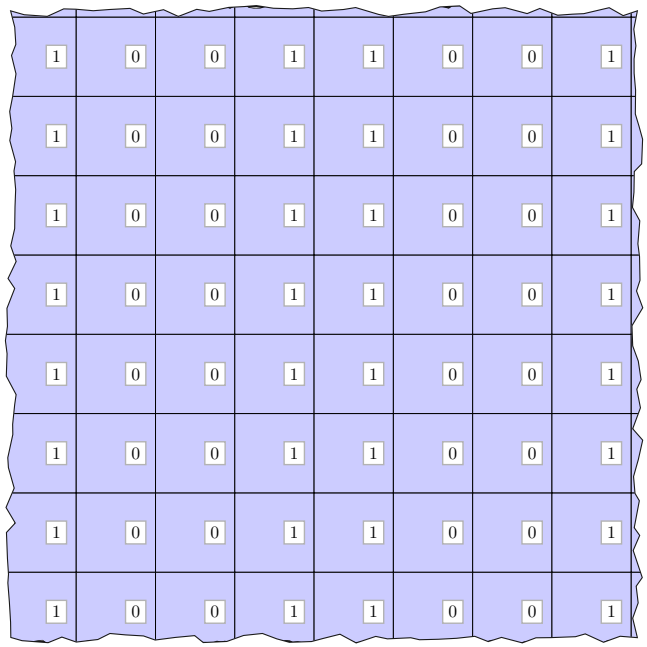


Quels ensembles peut-on réaliser ?

Théorème [Jeandel & V.] Pour tout langage $L \in \mathbf{NE}$, on peut construire un SFT X dont les périodes sont exactement L.

Remarque X n'est pas nécessairement de dimension 2.

Idée de la preuve. Soit L un langage de $\mathbf{NTIME}_1(n)$:

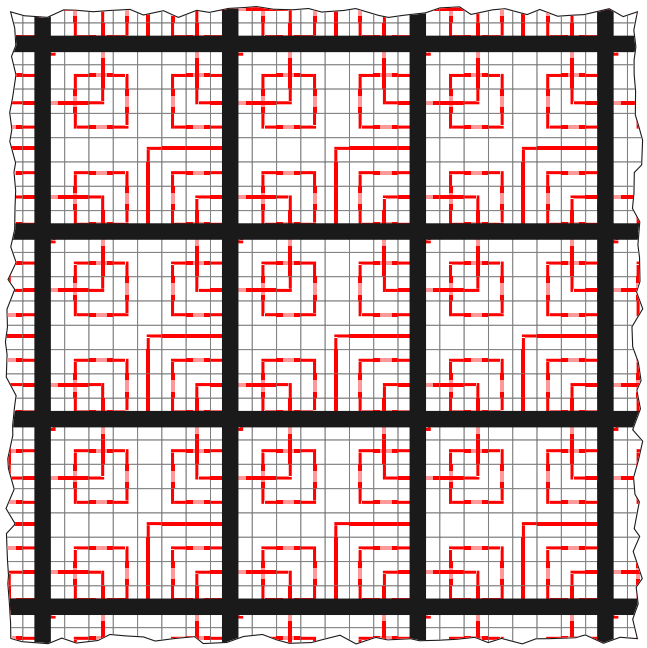


Quels ensembles peut-on réaliser ?

Théorème [Jeandel & V.] Pour tout langage $L \in \mathbf{NE}$, on peut construire un SFT X dont les périodes sont exactement L.

Remarque X n'est pas nécessairement de dimension 2.

Idée de la preuve. Soit L un langage de $\mathbf{NTIME}_1(n)$:

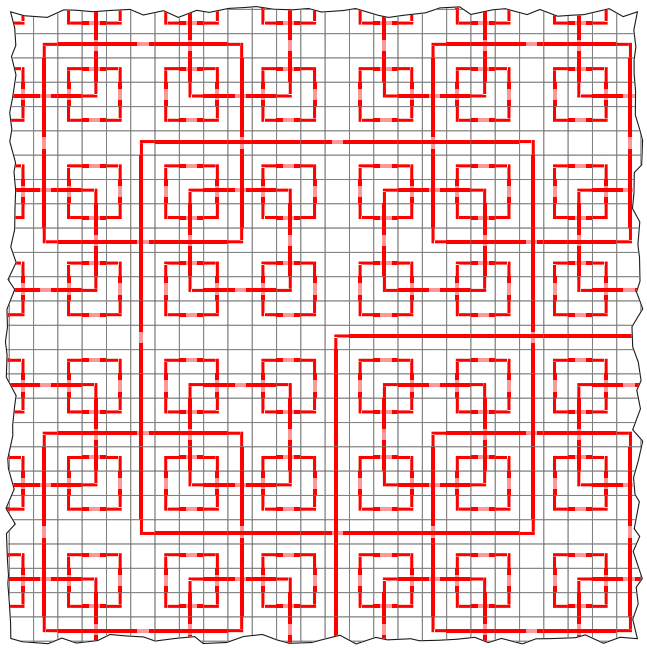


Quels ensembles peut-on réaliser ?

Théorème [Jeandel & V.] Pour tout langage $L \in \mathbf{NE}$, on peut construire un SFT X dont les périodes sont exactement L.

Remarque X n'est pas nécessairement de dimension 2.

Idée de la preuve. Soit L un langage de $\mathbf{NTIME}_1(n)$:



Quels ensembles peut-on réaliser ?

Théorème [Jeandel & V.] Pour tout langage $L \in \mathbf{NE}$, on peut construire un SFT X dont les périodes sont exactement L.

Remarque X n'est pas nécessairement de dimension 2.

Théorème [Jeandel & V.] Les ensembles de périodes sont exactement les langages de $\mathbf{NE}$.

Comptons !

Soit $\Omega_X(n)$ la fonction qui compte le **nombre de configurations de période n** de X.

Théorème Ω_X est un invariant de conjugaison.

Théorème [Jeandel & V.] Pour toute fonction $f : \mathbb{N}^* \to \mathbb{N}$,

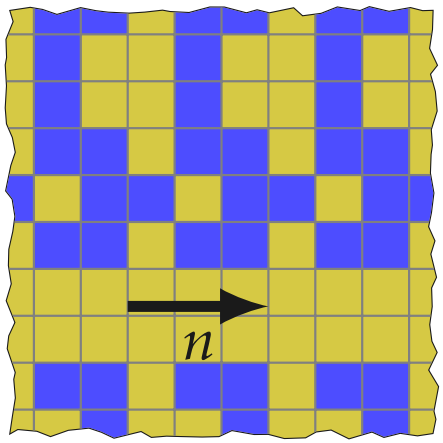
$$f \in \#\mathbf{E} \Leftrightarrow \text{ il existe un SFT } X \text{ tel que } \Omega_X = f$$

$\#\mathbf{E}$: Fonctions de comptage du nombre de chemins acceptants des machines de Turing de $\mathbf{NE}$.

Périodicité horizontale

On peut définir une autre notion de périodicité en dimension 2 :

Définition Une configuration est **périodique horizontalement** si il existe n tel que $\sigma_{ne_1}(x) = x$.



Théorème L'ensemble $\mathfrak{P}_X^h$ des périodes horizontales est invariant par conjugaison.

Théorème [Jeandel & V.] Les ensembles de périodes horizontales de SFTs sont **exactement** les langages de $\mathbf{NSPACE}_1(n)$.

1. Complexité algorithmique de la conjugaison et de la factorisation

2. Périodicité

3. Degrés Turing des SFTs

Degrés Turing

- $x \leq_T y$ s'il existe f récursive prenant y comme oracle et calculant x.
- $x \equiv_T y$ si $x \leq_T y$ et $x \geq_T y$.
- Un **degré Turing** est la classe d'équivalence pour la relation $\equiv_T$. On note $\deg_T x$ le degré de x.

Le degré le plus simple est $\mathbf{0}$: les suites **calculables**.

- Degré Turing d'une configuration.
- **Ensemble de degrés Turing** d'un sous-shift.

L'ensemble des degrés Turing d'un SFT est un invariant de conjugaison.

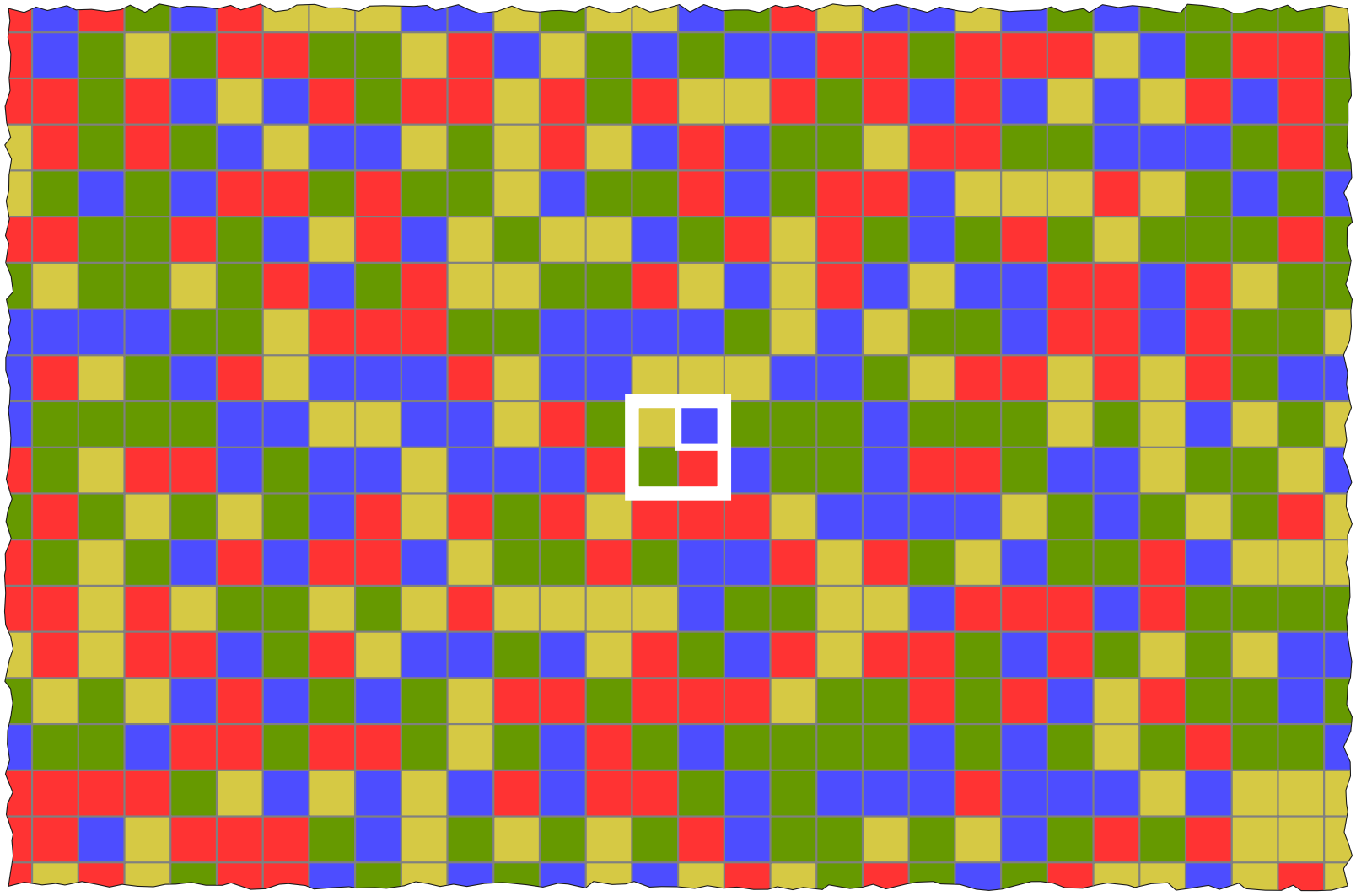
SFTs et Calculabilité

$S \subseteq \{0,1\}^{\mathbb{N}}$ est **effectivement clos** si c'est l'ensemble des oracles sur lesquels une machine de Turing M donnée ne s'arrête pas.

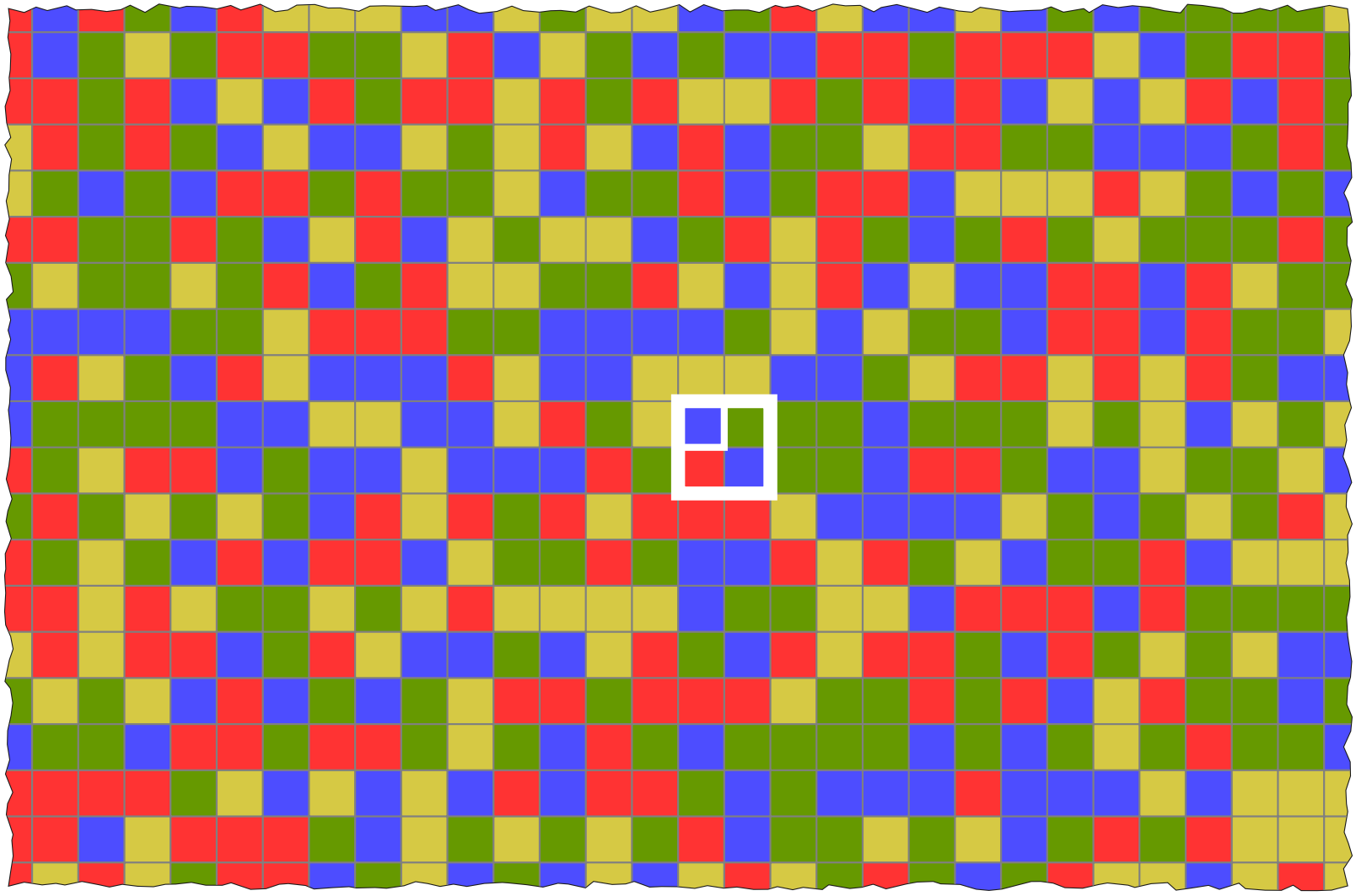
$$\forall s \in \{0,1\}^{\mathbb{N}}, \quad s \in S \Leftrightarrow M^s \uparrow$$

$\mathcal{X}_{\mathcal{F}}$ est un sous-ensemble effectivement clos de $\Sigma^{\mathbb{Z}^2}$:

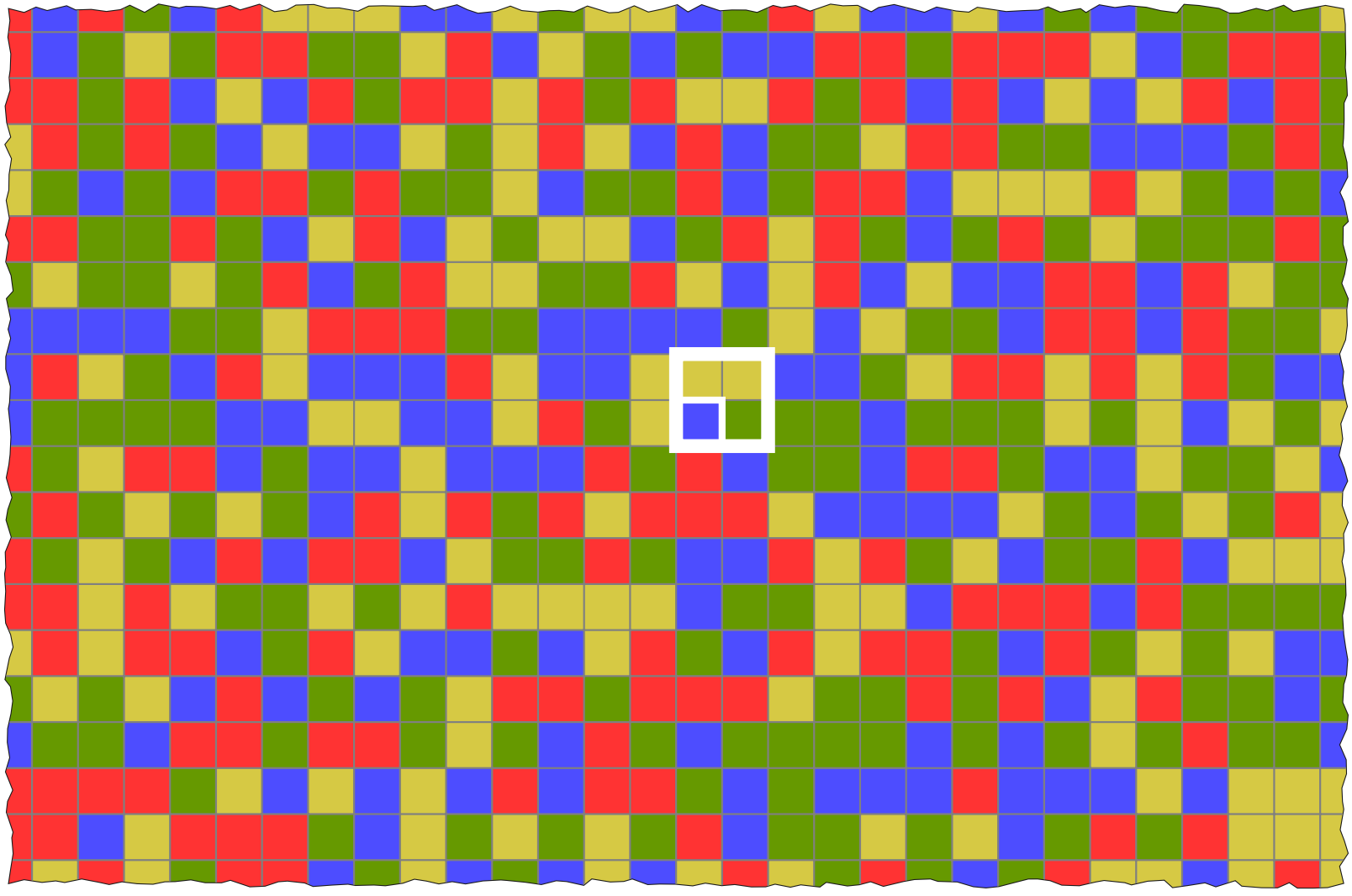
SFTs et Calculabilité



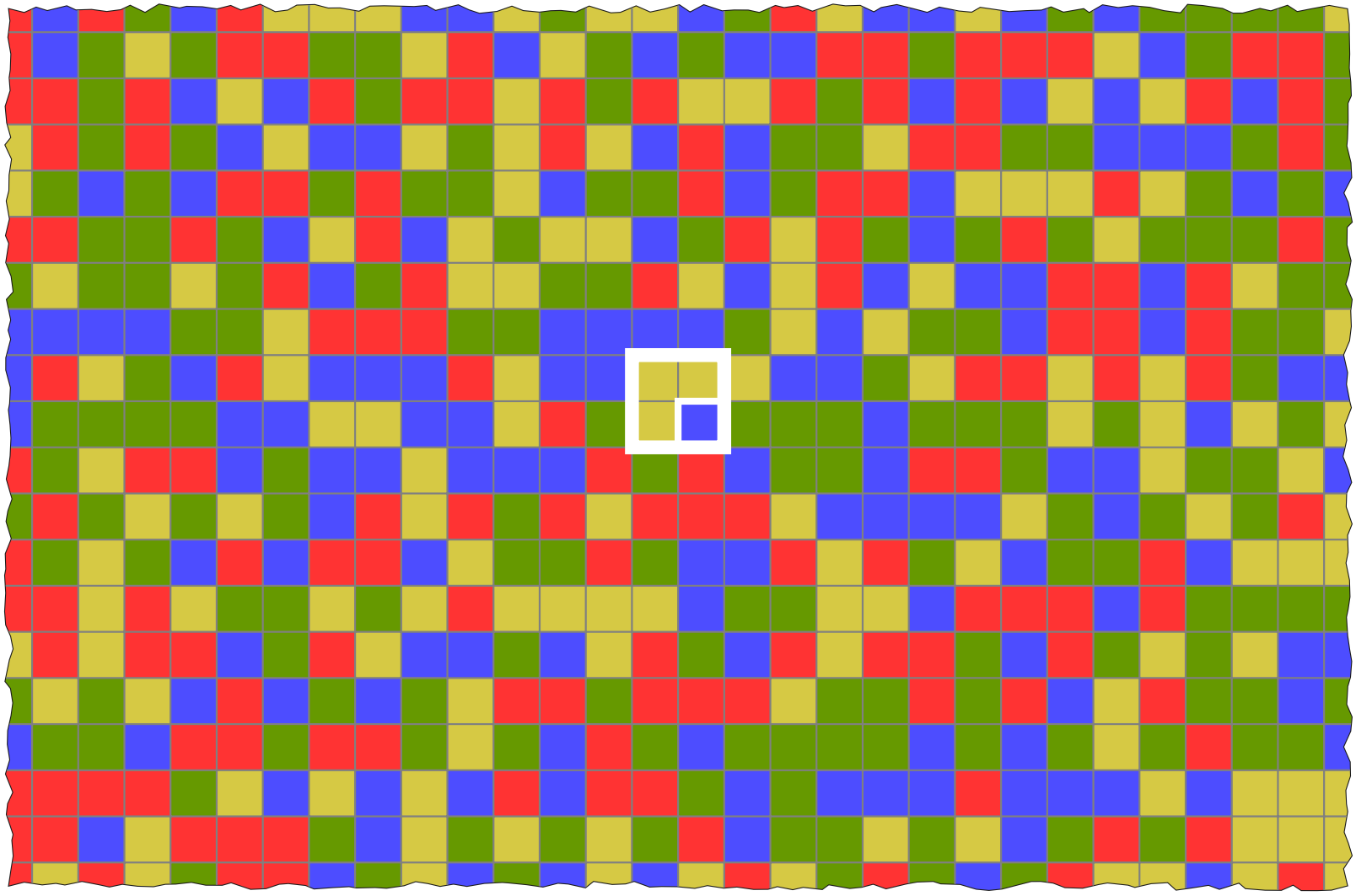
SFTs et Calculabilité



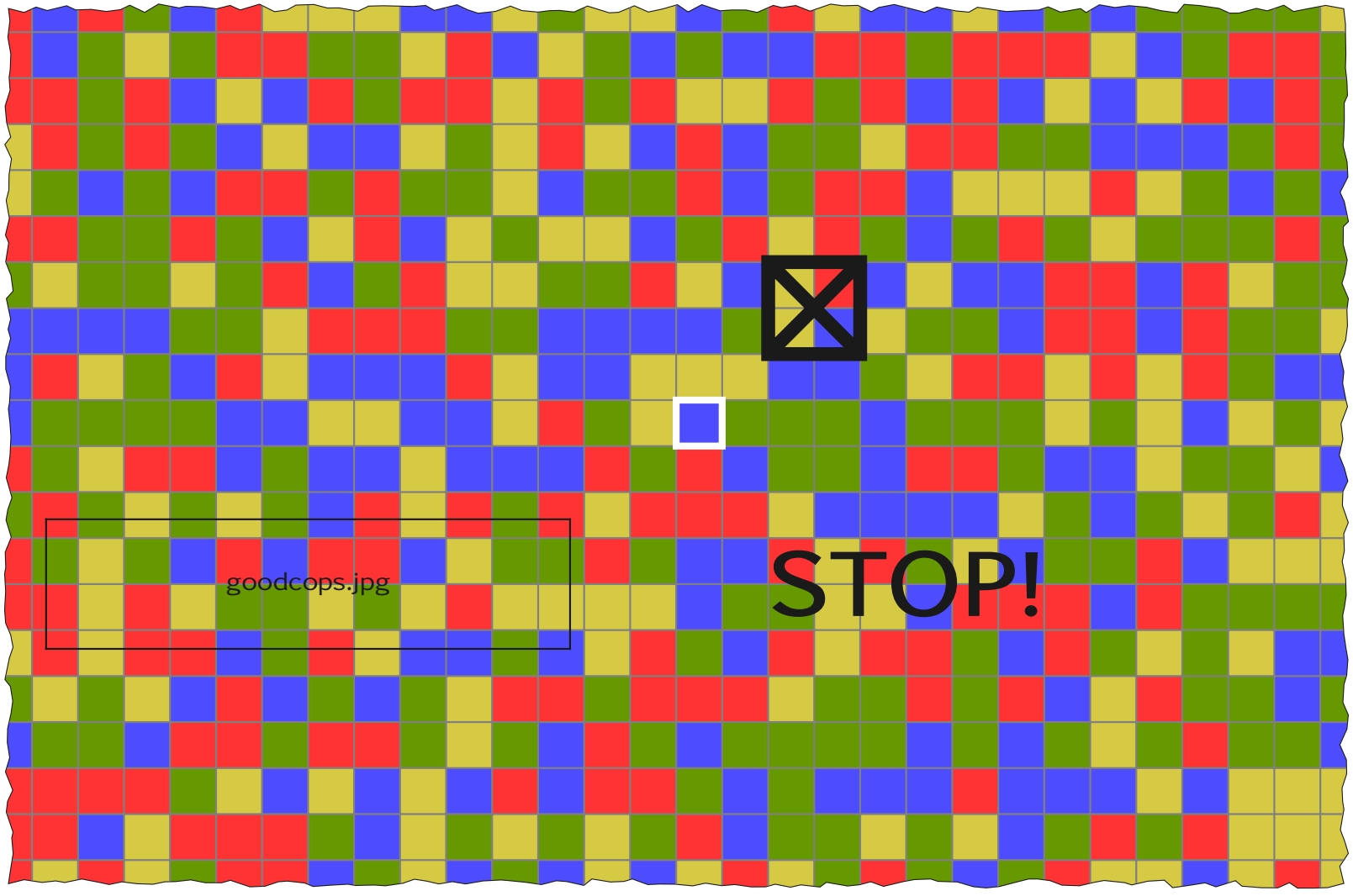
SFTs et Calculabilité



SFTs et Calculabilité



SFTs et Calculabilité



Degrés Turing des SFTs

Théorème [Jeandel & V.] Pour tout effectivement clos, il existe un SFT ayant le **même ensemble de degrés Turing, à 0 près**.

Degrés Turing des SFTs

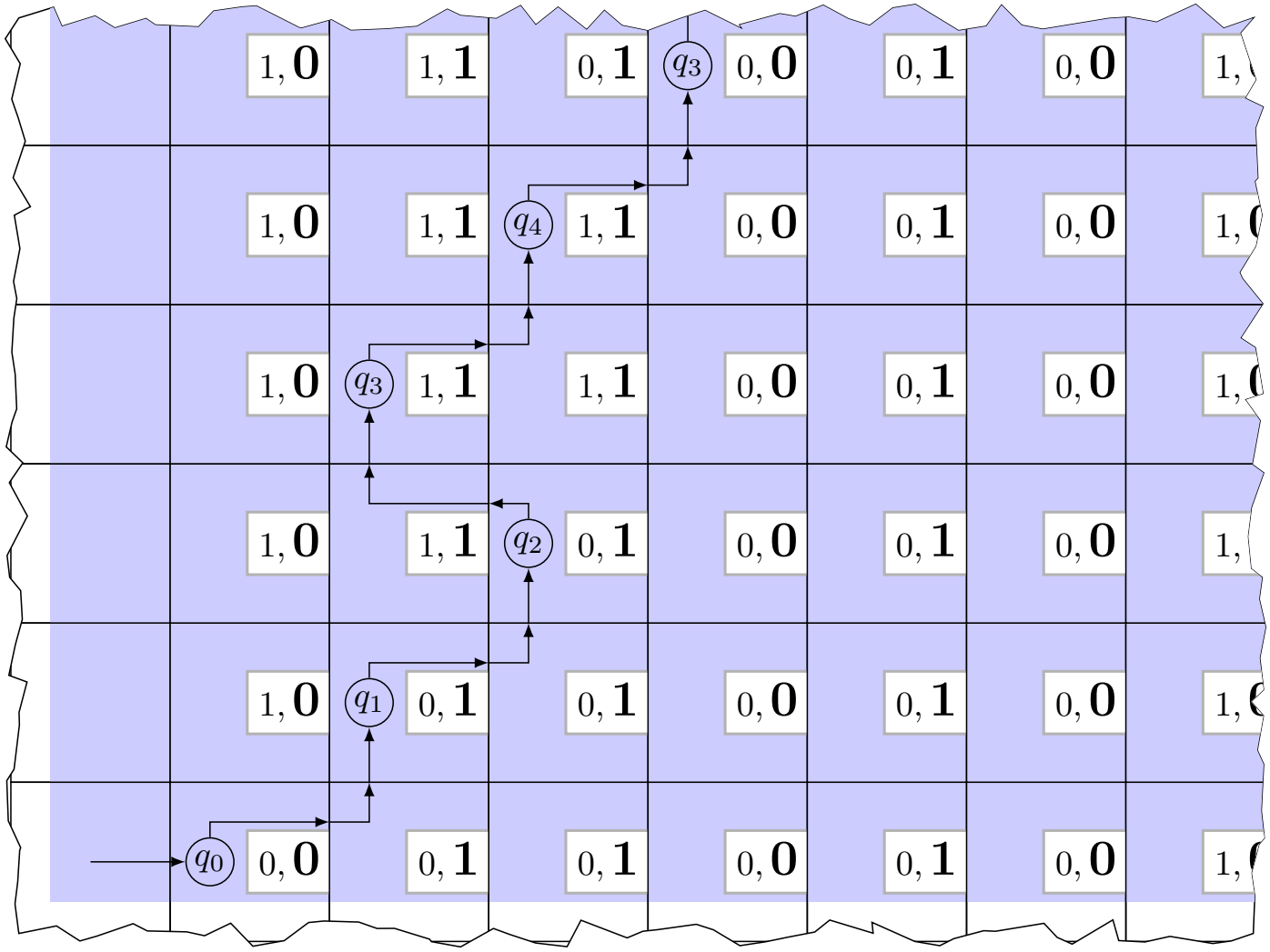
Théorème [Jeandel & V.] Pour tout effectivement clos, il existe un SFT ayant le **même ensemble de degrés Turing, à 0 près**.

Théorème [Jeandel & V.] Pour toute classe Π_1^0 S, il existe $\mathcal{F}$ et $C \subset \mathcal{X}_\mathcal{F}$ un **ensemble récursif de points récursifs** tel que :

$$(\mathcal{X}_\mathcal{F} \setminus C) \,/\mathbb{Z}^2 \cong S$$

- On quotiente par les translations.
- **Isomorphisme récursif:**
  Il existe une fonction bijective, récursive $f : S \to S'$, avec f^{-1} récursive.

Classes Π_1^0 et SFTs

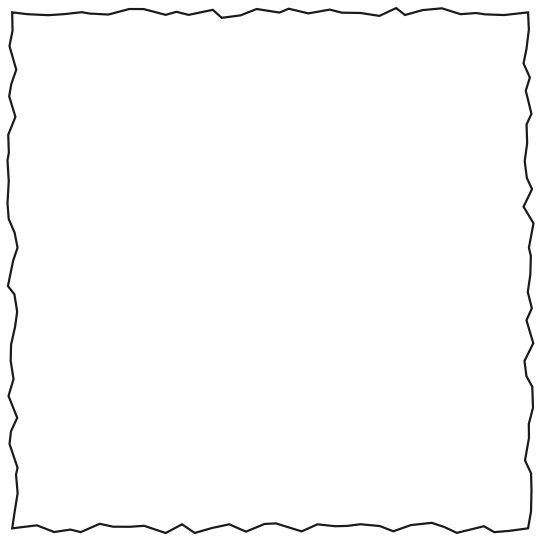


Il faut faire apparaître la tuile de début de calcul.

Preuve

- On a besoin d'une trame en forme de grille pour mettre du calcul.
- On a besoin d'un coin pour commencer le calcul.
- Les configurations sans calcul doivent être récursives (donc dénombrables).

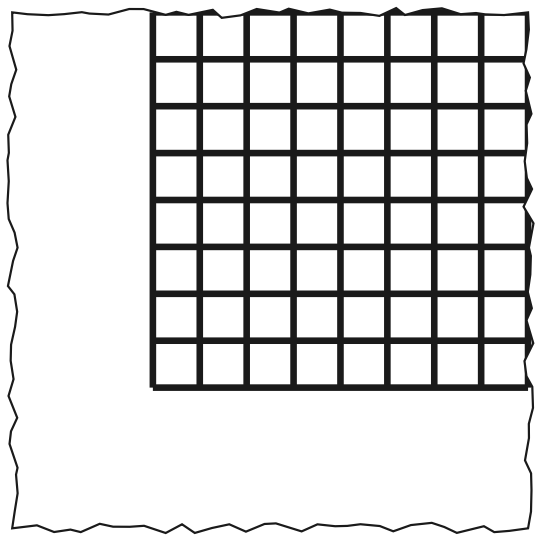
On ne peut pas réutiliser la construction de Robinson.



Preuve

- On a besoin d'une trame en forme de grille pour mettre du calcul.
- On a besoin d'un coin pour commencer le calcul.
- Les configurations sans calcul doivent être récursives (donc dénombrables).

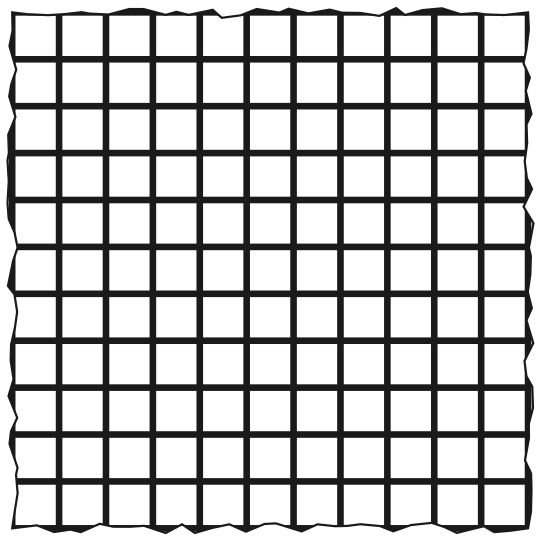
On ne peut pas réutiliser la construction de Robinson.



Preuve

- On a besoin d'une trame en forme de grille pour mettre du calcul.
- On a besoin d'un coin pour commencer le calcul.
- Les configurations sans calcul doivent être récursives (donc dénombrables).

On ne peut pas réutiliser la construction de Robinson.

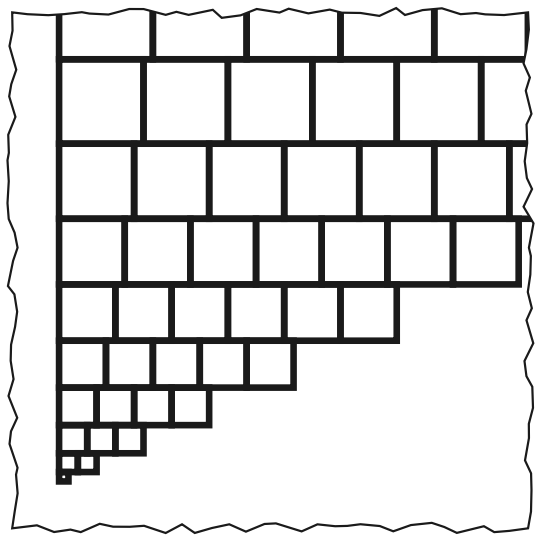


Problème. Certains points ne contiendront pas de calcul valide.

Preuve

- On a besoin d'une trame en forme de grille pour mettre du calcul.
- On a besoin d'un coin pour commencer le calcul.
- Les configurations sans calcul doivent être récursives (donc dénombrables).

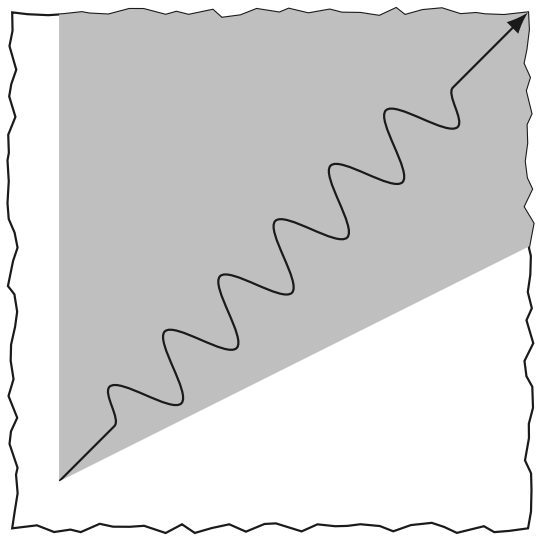
On ne peut pas réutiliser la construction de Robinson.



Preuve

- On a besoin d'une trame en forme de grille pour mettre du calcul.
- On a besoin d'un coin pour commencer le calcul.
- Les configurations sans calcul doivent être récursives (donc dénombrables).

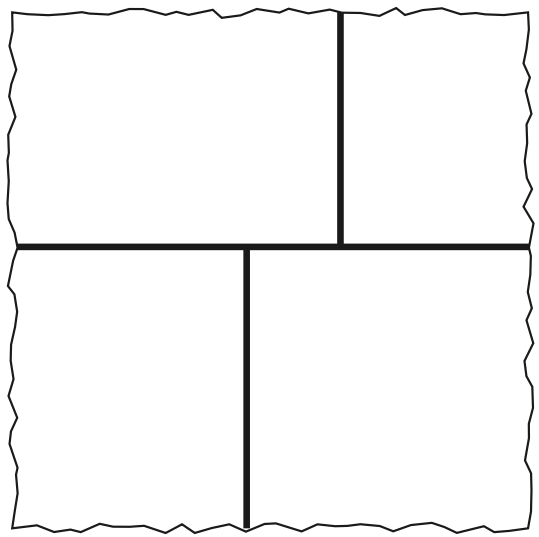
On ne peut pas réutiliser la construction de Robinson.



Preuve

- On a besoin d'une trame en forme de grille pour mettre du calcul.
- On a besoin d'un coin pour commencer le calcul.
- Les configurations sans calcul doivent être récursives (donc dénombrables).

On ne peut pas réutiliser la construction de Robinson.

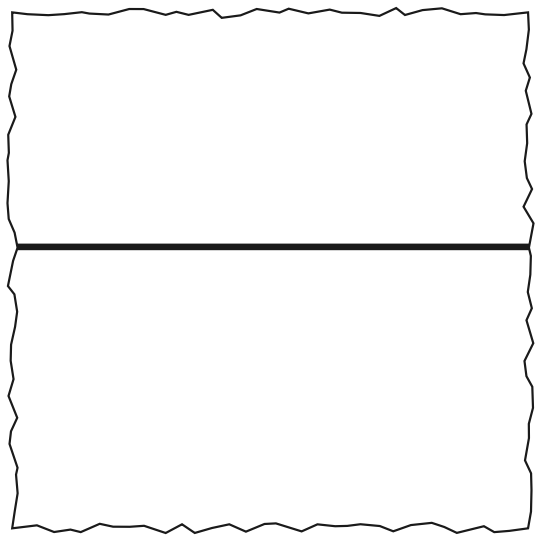


Configurations sans "vrai" calcul.

Preuve

- On a besoin d'une trame en forme de grille pour mettre du calcul.
- On a besoin d'un coin pour commencer le calcul.
- Les configurations sans calcul doivent être récursives (donc dénombrables).

On ne peut pas réutiliser la construction de Robinson.

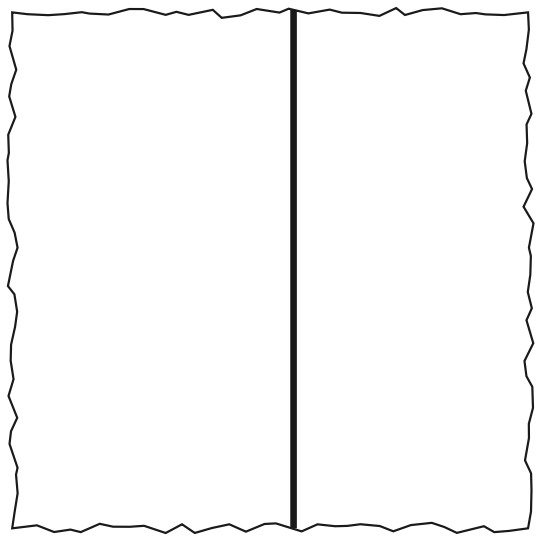


Configurations sans "vrai" calcul.

Preuve

- On a besoin d'une trame en forme de grille pour mettre du calcul.
- On a besoin d'un coin pour commencer le calcul.
- Les configurations sans calcul doivent être récursives (donc dénombrables).

On ne peut pas réutiliser la construction de Robinson.

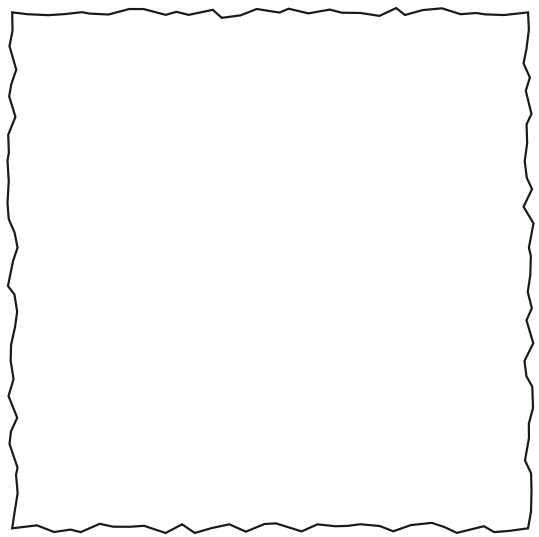


Configurations sans "vrai" calcul.

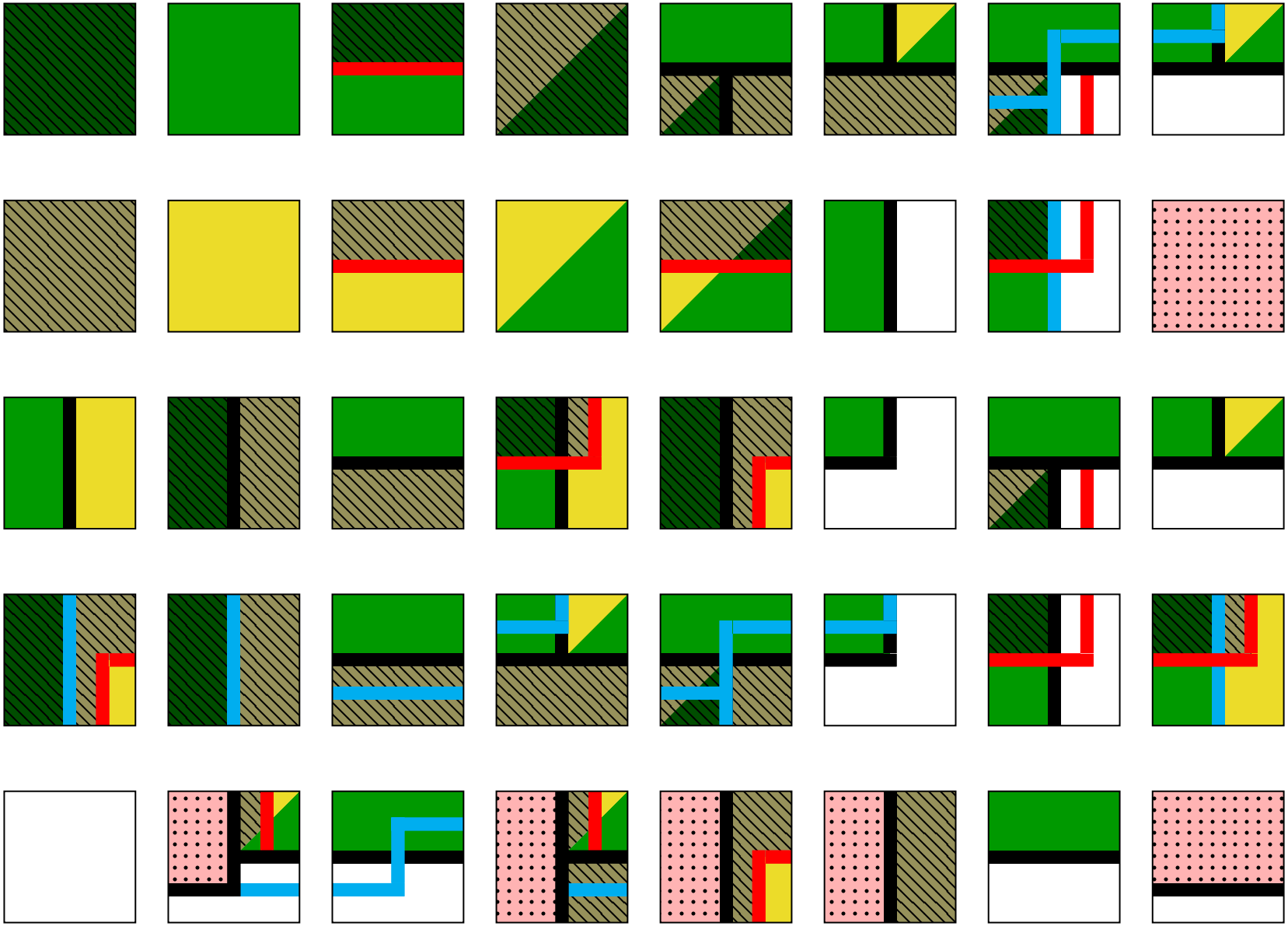
Preuve

- On a besoin d'une trame en forme de grille pour mettre du calcul.
- On a besoin d'un coin pour commencer le calcul.
- Les configurations sans calcul doivent être récursives (donc dénombrables).

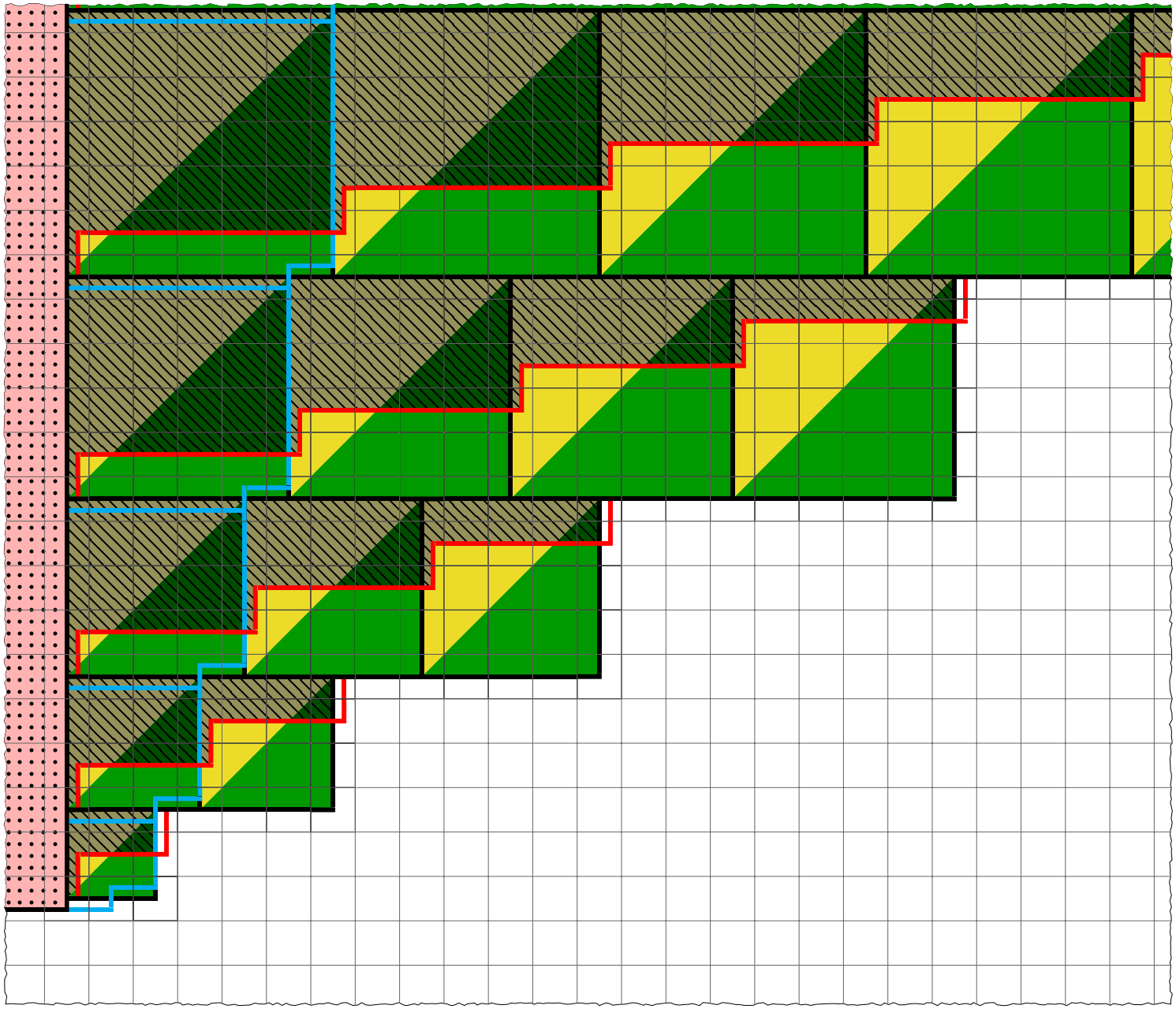
On ne peut pas réutiliser la construction de Robinson.



Configurations sans "vrai" calcul.



40 tuiles de Wang



Une seule configuration où mettre du calcul.

Degrés Turing

Théorème [Jeandel & V.] Pour tout ensemble effectivement clos S, il existe $\mathcal{F}$ et $C \subset \mathcal{X}_\mathcal{F}$ un **ensemble récursif de points récursifs** tel que :

$$(\mathcal{X}_\mathcal{F} \setminus C)\,/\mathbb{Z}^2 \cong S$$

Corollaire Pour tout ensemble effectivement clos, il existe un SFT ayant le **même ensemble de degrés Turing, à 0 près**.

Remarque Les SFTs ont quasiment les mêmes structures calculatoires que les ensembles effectivement clos.

Degrés Turing

Théorème [Jockusch & Soare 1972] Il existe des ensembles effectivement clos dont tous les **membres** sont **incomparables**.

Théorème [Jeandel & V.] Tout SFT ne contenant **aucune** configuration calculable contient un **cône de degrés Turing**.

Conséquence : Il existe des ensembles effectivement clos qui ne peuvent être récursivement isomorphes à aucun SFT.

Un **cône de degrés Turing** est un degré Turing et tous les degrés qui lui sont supérieurs.

Minimalité et degrés Turing

Définition Un **sous-shift minimal** est un sous-shift dont **toutes les configurations** ont les **mêmes motifs**.

Théorème [Jeandel & V.] Tout sous-shift minimal non périodique contient un cône de degrés Turing.

Théorème [Hochman & V.] Il existe des sous-shift minimaux avec deux cônes distincts.