
Stabilité de classes de sous-shifts par facteurs et sous-actions

Nathalie AUBRUN

Sous la direction de Mathieu SABLİK

Stage réalisé au CMI
LATP - Marseille

Table des matières

Introduction	3
1 Les sous-shifts : définitions et outils d'analyse	1
1.1 Définition topologique	1
1.2 Sous-shift défini par exclusion de motifs	1
1.2.1 Définition et exemple	1
1.2.2 Sous-shifts de type fini	2
1.2.3 Cas particuliers de SFT : automates cellulaires et machines de Turing	3
1.3 Langage d'un sous-shift	4
1.3.1 Définitions	4
1.3.2 Propriétés du langage d'un sous-shift	4
1.4 Sous-shift défini par substitution	5
1.4.1 Définition et lien avec les SFT	5
1.4.2 Exemple de la substitution s_p	6
1.4.3 Construction d'espaces de calcul	7
2 Sous-shifts, sous-actions et facteurs	9
2.1 Facteur d'un sous-shift	9
2.1.1 Définition et exemples	9
2.1.2 Sous-shift sofique	9
2.2 Sous-action	9
2.2.1 Définition	10
2.2.2 Sous-shifts récursivement énumérables	12
3 Clôture de classes des sous-shifts	15
3.1 Opérations et clôtures	15
3.2 Traduction de résultats en terme de clôture	16
3.3 Un ordre sur les langages	16
3.4 Théorème de clôture	17
3.4.1 Inclusion directe	17
3.4.2 Inclusion réciproque	17
Conclusion	21

Introduction

Les sous-shifts sont des objets utilisés pour modéliser les systèmes dynamiques discrets. Lors de ce stage je me suis intéressée aux sous-shifts multi-dimensionnels qui sont des objets encore mal compris.

Un système dynamique discret (X, F) consiste en un ensemble de configurations X (appelé l'espace des phases) dont l'évolution au cours du temps est donnée par une fonction $F : X \rightarrow X$. Chaque configuration initiale $x_0 \in X$ décrit une trajectoire $(x_n)_{n \in \mathbb{N}}$ dans l'espace, où $x_n = F^n(x_0)$ pour tout $n \in \mathbb{N}$. Pour étudier ces trajectoires, il est pertinent pour simplifier le problème de discrétiser l'espace en se donnant un recouvrement fini de fermés $X = \bigcup_{i=1}^n A_i$. En notant \mathcal{A} l'alphabet fini $\{A_1, \dots, A_n\}$, on s'intéresse alors à l'ensemble $\Sigma = \{(x_n) \in \mathcal{A}^{\mathbb{N}} / \exists x \in X, \forall n \in \mathbb{N}, F^n(x) \in A_i \Rightarrow x_n = A_i\}$ qui possède les propriétés topologiques intéressantes : c'est un ensemble fermé et stable par le décalage σ .

Ces deux propriétés caractérisent complètement ce type d'ensembles, qu'on appelle des sous-shifts. L'étude de tels systèmes a donné naissance à un nouveau domaine : la dynamique symbolique. Par analogie avec ce qui précède et qui concerne la dimension 1, en dimension supérieure on parle de sous-shift multi-dimensionnel pour une partie fermée et stable par décalage de $\mathcal{A}^{\mathbb{Z}^d}$, où \mathcal{A} est un alphabet fini. Les sous-shifts ont ainsi à l'origine une définition topologique mais il existe d'autres définitions équivalentes plus combinatoires. Une manière naturelle de définir un sous-shift est de donner son langage, c'est-à-dire l'ensemble des motifs qui apparaissent dans ses configurations. Cette définition possède l'avantage de donner la liste exhaustive des motifs autorisés ; en contrepartie un tel ensemble de mots doit répondre à certaines exigences (être prolongeable et stable par sous-motif) ce qui le rend difficile à exploiter. Comme nous le verrons une autre définition possible, plus utilisable dans la pratique, est celle par exclusion de motifs. Si F est un ensemble de motifs le sous-shift associé est :

$$X_F = \{x \in \mathcal{A}^{\mathbb{Z}^d} / \forall m \in F, m \text{ n'apparaît pas dans } x\}$$

N'importe quel sous-shift peut être défini de cette manière ; en imposant à l'ensemble des motifs interdits F d'être fini, on se restreint à la classe des sous-shifts de type fini (SFT).

En dimension 1, la classe des SFT a beaucoup été étudiée et est très bien comprise, notamment grâce à certains outils de l'informatique théorique. On sait par exemple que le problème du sous-shift vide $X_F = \emptyset?$ pour un SFT est décidable [LM95]. De plus les SFT sont exactement les sous-shifts dont le langage est reconnu par automate local [Bea93]. Ce point de vue appelle d'autres questions, comme par exemple chercher à caractériser de manière simple la classe définie par un reconnaiseur de langage (automate fini, automate à pile, machine de Turing, ...). Dans le cas des automates finis cette classe est connue, c'est la classe des sous-shifts sofiques que l'on obtient par facteur d'un SFT [LM95].

Dans le cas multi-dimensionnel les choses se compliquent. En dimension 2 un SFT peut être vu comme un pavage, et le problème $X_F = \emptyset?$ devient alors indécidable [Ber66] puis [Rob71] ; de plus on ne sait pas définir l'équivalent des automates finis sur des motifs à plusieurs dimensions. Cependant les récents travaux d'Hochman [HM07] et [Hoc07b] ouvrent de nouvelles perspectives en étudiant en plus des facteurs une nouvelle opération dynamique propre à la dimension supérieure, la sous-action.

Après avoir rappelé dans une première partie les différentes définitions des sous-shifts et exposé quelques unes de leurs propriétés, on étudiera en détail dans la deuxième partie du rapport ces deux opérations sur les SFT. A travers des contre-exemples on verra notamment que la classe des SFT n'est stable par aucune de ces opérations, mais que l'on peut exhiber des classes stables les contenant : celles des sofiques pour le facteur et celle des sous-shifts récursivement énumérables (dont l'ensemble des motifs interdits est un langage récursivement énumérable) pour la sous-action.

Hochman montre [Hoc07b] que tout sous-shift RE de dimension d s'obtient à partir d'un SFT de dimension $d + 2$ à l'aide de facteurs et de sous-actions. Derrière ce résultat apparaît l'idée de simulation d'un sous-shift par un autre, dans laquelle la notion de simulation dépend des opérations que l'on autorise. Car en plus des facteurs et sous-action, il existe d'autres opérations dynamiques tout aussi naturelles : produit, type fini (intersection avec un SFT), superposition (pour augmenter la dimension) et injection.

La troisième partie du rapport est consacrée à l'étude des clôtures de certaines classes par ces opérations. A partir d'un sous-shift initial, on cherche à déterminer les sous-shifts obtenus en appliquant un nombre fini d'opérations choisies parmi celles citées précédemment. Nous verrons que certains résultats déjà connus s'expriment de manière simple avec ce formalisme. La clôture permet aussi de définir un

ordre sur les sous-shifts : on note $\Sigma \leq_{op} \Sigma'$ si on peut transformer Σ' en Σ en un nombre fini d'opérations contenues dans *op*. Comme on le verra, l'ensemble des diagrammes espace-temps générés par un automate cellulaire est un SFT bi-dimensionnel. Dans ce cas l'utilisation de cette notion de simulation, pour certaines opérations, correspond à la notion de groupage introduite par [MR99, Oll02, The05] pour l'étude des automates cellulaires.

Le principal résultat de ce rapport établit une correspondance entre l'ordre sur les sous-shifts donné par les opérations de produit, facteur, sous-action, superposition et type fini et un ordre \preceq sur les langages défini par les machines de Turing à semi-oracle (version modifiée des machines de Turing à oracle). Plus précisément, on montre les sous-shifts obtenus pour les opérations précédentes appliquées à un sous-shift $X_{\mathcal{L}}$ sont les sous-shifts $X_{\mathcal{L}'}$ tels que $\mathcal{L}' \preceq \mathcal{L}$.

1 Les sous-shifts : définitions et outils d'analyse

1.1 Définition topologique

Soit \mathcal{A} un alphabet fini. On s'intéresse ici à des coloriage du réseau \mathbb{Z}^d par l'ensemble \mathcal{A} , aussi appelés configurations. L'ensemble $\mathcal{A}^{\mathbb{Z}^d}$ des configurations, muni de la topologie produit est compact, métrisable et totalement discontinu.

On définit le **décalage** σ comme l'action naturelle de \mathbb{Z}^d sur l'espace des configurations, où un vecteur $u \in \mathbb{Z}^d$ agit sur $\mathcal{A}^{\mathbb{Z}^d}$ par décalage via le morphisme $\sigma_u : \mathcal{A}^{\mathbb{Z}^d} \rightarrow \mathcal{A}^{\mathbb{Z}^d}$ défini par :

$$\sigma_u(x)_i = x_{i+u} \text{ pour tout } i \in \mathbb{Z}^d.$$

Si $x \in \mathcal{A}^{\mathbb{Z}^d}$ est une configuration et $\mathbb{S} \subseteq \mathbb{Z}^d$ un ensemble de coordonnées, la restriction de x à \mathbb{S} notée $x_{\mathbb{S}}$ est un élément de $\mathcal{A}^{\mathbb{S}}$ tel que :

$$\forall u \in \mathbb{S}, (x_{\mathbb{S}})_u = x_u.$$

On munit $\mathcal{A}^{\mathbb{Z}^d}$ de la distance suivante :

$$d(x, y) = 2^{-\min\{n \in \mathbb{N} / x_{[-n, n]^d} \neq y_{[-n, n]^d}\}}$$

avec laquelle on définit une topologie équivalente à la topologie produit.

Si \mathbb{S} est un sous-ensemble fini de \mathbb{Z}^d et $m \in \mathcal{A}^{\mathbb{S}}$, on appelle **cylindre** relatif à m l'ensemble :

$$[m] = \{x \in \mathcal{A}^{\mathbb{Z}^d} / x_{\mathbb{S}} = m\}.$$

Les cylindres sont des ouverts-fermés.

Un **sous-shift** est un sous-ensemble fermé et stable par le décalage σ de $\mathcal{A}^{\mathbb{Z}^d}$. Il s'agit d'une définition topologique, nous allons voir qu'il en existe d'autres plus combinatoires.

1.2 Sous-shift défini par exclusion de motifs

On donne ici une définition équivalente qui sera celle qu'on utilisera le plus dans la suite. Un sous-shift peut être vu comme un ensemble de configurations dans lesquelles n'apparaissent pas certains motifs interdits. Un cas particulier important est celui des sous-shifts de type fini, lorsque l'ensemble des motifs interdits est fini.

1.2.1 Définition et exemple

Si $\mathbb{S} \subseteq \mathbb{Z}^d$ est un ensemble fini, on dit que $m \in \mathcal{A}^{\mathbb{S}}$ est un motif et \mathbb{S} est le support du motif m , noté $\text{supp}(m)$. On dit qu'un motif m **apparaît** dans $x \in \mathcal{A}^{\mathbb{Z}^d}$ s'il existe $i \in \mathbb{Z}^d$ tel que $x_{i+\mathbb{S}} = m$ et on note $m \sqsubseteq x$. De la même façon si m et m' sont deux motifs, on dit que m est un **sous-motif** de m' si $\text{supp}(m) \subset \text{supp}(m')$ et $\forall i \in \text{supp}(m), m_i = m'_i$ et on note $m \sqsubseteq m'$.

On aura besoin de la suite de la notion d'inclusion stricte : on dit qu'un ensemble fini $\mathbb{I} \subseteq \mathbb{Z}^d$ prolonge un ensemble fini $\mathbb{J} \subseteq \mathbb{Z}^d$ dans toutes les directions si $\forall u \in \mathbb{J}, \forall i \in \{1, \dots, d\} \exists u', u'' \in \mathbb{I}$ tels que $u' = u + e_i$ et $u'' = u - e_i$, où e_i est le $i^{\text{ème}}$ vecteur de la base canonique de \mathbb{Z}^d . On note $m \sqsubset m'$ si $m \sqsubseteq m'$ et que $\text{supp}(m')$ prolonge $\text{supp}(m)$ dans toutes les directions.

Si F est un ensemble de motifs, on définit le sous-shift associé à F par

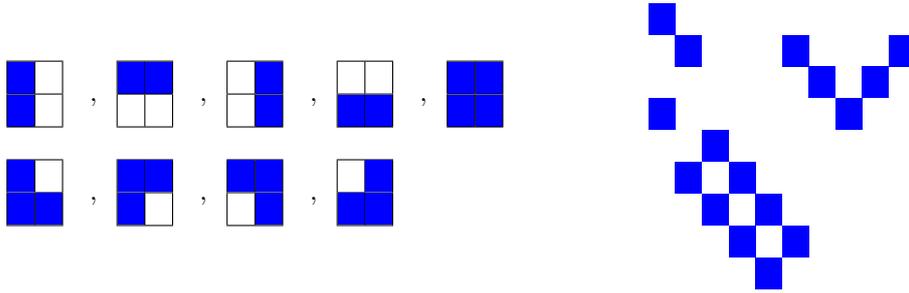
$$X_F = \{x \in \mathcal{A}^{\mathbb{Z}^d} / \forall m \in F, m \text{ n'apparaît pas dans } x\}.$$

En écrivant cette définition à l'aide des cylindres on a :

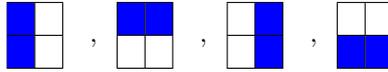
$$\Sigma_F = \bigcap_{m \in F} \bigcap_{u \in \mathbb{Z}^d} \sigma^u([m_i]^c) = \bigcap_{m \in F} \bigcap_{u \in \mathbb{Z}^d} (\sigma^u([m_i]))^c = \mathcal{A}^{\mathbb{Z}^d} \setminus \bigcup_{m \in F} \bigcup_{u \in \mathbb{Z}^d} \sigma^u([m_i])$$

Cette écriture permet de retrouver facilement la définition topologique : la stabilité par le décalage σ apparaît clairement, et l'ensemble Σ_F est fermé comme intersection de fermés.

Exemple Sur l'alphabet $\mathcal{A} = \{\square, \blacksquare\}$, un exemple d'ensemble de motifs interdits et une configuration du sous-shift ainsi défini :



Remarque L'ensemble F n'est pas unique. Dans l'exemple précédent il suffisait d'interdire :



pour obtenir le même sous-shift. Il faudra avoir cette remarque en tête lorsqu'on définira des sous-shifts avec des propriétés particulière sur l'ensemble des motifs interdits.

1.2.2 Sous-shifts de type fini

On peut imposer des conditions à l'ensemble des motifs interdits pour définir des classes de sous-shifts. On dit qu'un sous-shift Σ est **de type fini** (SFT) s'il existe un ensemble fini de motifs F tel que $\Sigma = X_F$.

Pour simplifier les raisonnements on remarquera les propriétés suivantes :

- on peut choisir tous les motifs interdits de même support élémentaire ;
- il est équivalent de se donner un ensemble fini de motifs autorisés ;
- il est équivalent de se donner un ensemble fini de tuiles et un ensemble de règles d'adjacence sur ces tuiles ;
- si $\Sigma = X_F \subseteq \mathcal{A}^{\mathbb{Z}^d}$ et $\Sigma' = X_{F'} \subseteq \mathcal{B}^{\mathbb{Z}^d}$ sont deux SFT de même dimension, alors $X_{F \times F'} \subseteq (\mathcal{A} \times \mathcal{B})^{\mathbb{Z}^d}$ est aussi un SFT.

Cette dernière propriété peut être vue comme un résultat de stabilité de la classe des SFT par l'opération produit. On reviendra sur cette opération dans la troisième partie.

Les SFT joueront dans la suite un rôle important. D'abord parce que leur description par exclusion de motifs est simple, on peut facilement construire des exemples même en dimension 2. Ensuite ce sont des objets très bien compris en dimension 1, on peut donc s'inspirer de résultats déjà connus pour essayer de les étendre à la dimension supérieure.

On verra aussi qu'il permette de faire le lien avec d'autres domaines d'étude, comme celui des automates cellulaires.

1.2.3 Cas particuliers de SFT : automates cellulaires et machines de Turing

Une machine de Turing est un modèle de calcul que l'on peut facilement simuler avec un SFT de dimension 2 : une dimension représente l'espace de travail et la seconde le temps de calcul. De la même façon un automate cellulaire de dimension d peut être simulé par un SFT de dimension $d + 1$.

Automates cellulaires Soit \mathcal{Q} un ensemble d'états et $\mathbb{V} \subset \mathbb{Z}^d$ un ensemble fini (voisinage de la fonction de transition). Un automate cellulaire est la donnée d'une fonction locale $f : \mathcal{Q}^{\mathbb{V}} \rightarrow \mathcal{Q}$ à partir de laquelle on construit une fonction globale $F : \mathcal{Q}^{\mathbb{Z}^d} \rightarrow \mathcal{Q}^{\mathbb{Z}^d}$ définie par :

$$\forall x \in \mathcal{Q}^{\mathbb{Z}^d}, \forall u \in \mathbb{Z}^d, (F(x))_u = f(x_{u+\mathbb{V}})$$

On peut représenter un automate cellulaire par l'ensemble fini des règles locales qui le définissent :

$$R = \{(m, y) \in \mathcal{Q}^{\mathbb{V}} \times \mathcal{Q} / f(m) = y\}$$

Le sous-shift $\Sigma = X_{\mathcal{Q}^{\mathbb{V}} \times \mathcal{Q} \setminus R} \subseteq \mathcal{Q}^{\mathbb{Z}^d \times \mathbb{N}}$ est l'ensemble des diagrammes espace-temps de l'automate cellulaire F .

Remarque Ici on définit un sous-shift non pas sur \mathbb{Z}^{d+1} mais sur un demi-espace $\mathbb{Z}^d \times \mathbb{N}$ puisque le temps ne s'écoule que dans un sens. Pour définir un automate cellulaire comme un sous-shift sur l'espace tout entier, on a deux solutions :

- se limiter aux automates surjectifs. Ainsi chaque configuration possède au moins un antécédent par F , et le SFT décrit précédemment correspond à l'ensemble des diagrammes espace-temps. Inconvénient : tous les automates cellulaires ne sont pas surjectifs.
- interdire le même ensemble de motifs que sur $\mathbb{Z}^d \times \mathbb{N}$. On obtient tous les diagrammes espace-temps des configurations de l'ensemble limite de l'automate : $\Lambda_F = \bigcap_{n \in \mathbb{N}} F^n(\mathcal{A}^{\mathbb{Z}^d})$, qui est le plus grand ensemble sur lequel F est surjective. Inconvénient : on perd tous les diagrammes espace-temps des configurations qui ne sont pas dans l'ensemble limite.

Question : Quelles propriétés vérifiées par les automates cellulaires le sont aussi par les SFT ?

Machines de Turing Une machine de Turing est un septuplet $(Q, \Sigma, \Gamma, \#, q_0, \delta, F)$ où :

- Q est un ensemble fini d'états ; $q_0 \in Q$ est l'état initial
- Σ et Γ sont deux alphabets finis avec $\Sigma \subsetneq \Gamma$
- $\# \notin \Gamma$ est le symbole blanc
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{\leftarrow, \cdot, \rightarrow\}$ est la fonction de transition
- $F \subset Q$ est l'ensemble des états finaux.

De même que pour les automates cellulaires, on va coder la machine de Turing par son ensemble de règle. Par exemple la règle $\delta(q_1, x) = (q_2, y, \leftarrow)$ sera codée par le motif :

(q_2, z)	y	z'
z	(q_1, x)	z'

et si q_f est un état final, alors la machine arrête son calcul ce que l'on traduit par le motif :

z	(q_f, x)	z'
z	(q_f, x)	z'

Avec cette construction on n'est pas sûr que les calculs que l'on simule soient effectivement des calculs de la machine de Turing, tout simplement parce qu'il peut y avoir plusieurs têtes de lecture en parallèle. On peut aussi imaginer des configurations qui représente un calcul sur un mot d'entrée infini, ou tout simplement une configuration qu'il est impossible de faire apparaître avec un calcul sur une entrée valide.

Pour remédier à ces problèmes, il suffit de donner un cadre de calcul à la machine. On peut par exemple utiliser un second SFT permettant de construire des rectangles, et utiliser l'opération produit pour le superposer au sous-shift défini par les règles de la machine de Turing. On ajoute ensuite des motifs interdits pour placer une tête de lecture sur le coin inférieur gauche de chaque rectangle, et interdire les têtes de lecture sur le reste de la base.

Proposition 2 Soit $\mathcal{L} \subset \mathcal{E}^d$. \mathcal{L} est le langage d'un sous-shift si et seulement si \mathcal{L} est prolongeable et stable par sous-mot élémentaire.

Démonstration

- Soit $\mathcal{L} = \mathcal{L}(\Sigma)$ le langage d'un sous-shift. Si $m \in \mathcal{L}$ et $m' \sqsubseteq m$, alors il existe $x \in \Sigma$ tel que $m \sqsubseteq x$, et donc a fortiori $m' \sqsubseteq x$. Ainsi $m' \in \mathcal{L}$, c'est-à-dire que \mathcal{L} est stable par sous-mot élémentaire. D'autre part si $m \in \mathcal{L}$ tel que $m \in \mathcal{L}_n(\Sigma)$, il existe $x \in \Sigma$ tel que $x|_{[-n;n]^d} = m$. On pose $m' = x|_{[-(n+1);n+1]^d}$. Alors $m \sqsubset m'$ et $m' \in \mathcal{L}$ donc \mathcal{L} est prolongeable.
- Soit $\mathcal{L} \subset \mathcal{E}$ un langage prolongeable et stable par sous-mot élémentaire. On considère $\Sigma = X_{\mathcal{L}^c}$; montrons que $\mathcal{L} = \mathcal{L}(\Sigma)$.
 - Si $m \in \mathcal{L}(\Sigma)$, alors $m \sqsubseteq x \in X_{\mathcal{L}^c}$ donc $m \notin \mathcal{L}^c$ i.e. $m \in \mathcal{L}$. On a bien $\mathcal{L}(\Sigma) \subseteq \mathcal{L}$.
 - Si $m \in \mathcal{L}$, alors il existe un entier n tel que $m \in \mathcal{A}^{[-n;n]^d}$. Comme le langage \mathcal{L} est prolongeable, il existe $m_1 \in \mathcal{A}^{[-(n+1);n+1]^d} \cap \mathcal{L}$ tel que $m \sqsubset m_1$. En itérant le procédé, on construit une suite $(m_k)_{k \in \mathbb{N}}$ d'éléments de \mathcal{L} telle que $m \sqsubset m_1 \sqsubset \dots \sqsubset m_k \sqsubset m_{k+1} \sqsubset \dots$. Notons x_k un élément de $\mathcal{A}^{\mathbb{Z}^d}$ tel que $(x_k)|_{\mathbb{S}_{n+k}} = m_k$. Par compacité il existe ϕ une extraction telle que $(x_{\phi(k)})_{k \in \mathbb{N}}$ converge; on appelle $x = \lim x_{\phi(k)}$, montrons que $x \in \Sigma$. Soit m' un motif qui apparaît dans x . Alors $\exists k \in \mathbb{N}$ tel que $m' \sqsubseteq m_k$, donc $m' \notin \mathcal{L}^c$. On a bien $x \in \Sigma$ donc m est autorisé dans Σ c'est-à-dire $\mathcal{L} \subseteq \mathcal{L}(\Sigma)$. \square

1.4 Sous-shift défini par substitution

Dans cette partie on se limite au cas de la dimension 2. Une manière simple de construire un ensemble de motifs est de partir des lettres de l'alphabet \mathcal{A} (motifs élémentaires les plus simples) et de remplacer chaque lettre par un motif fixé non linéaire. Une telle transformation s'appelle une substitution et permet par itération de construire des motifs aussi grands qu'on le souhaite.

1.4.1 Définition et lien avec les SFT

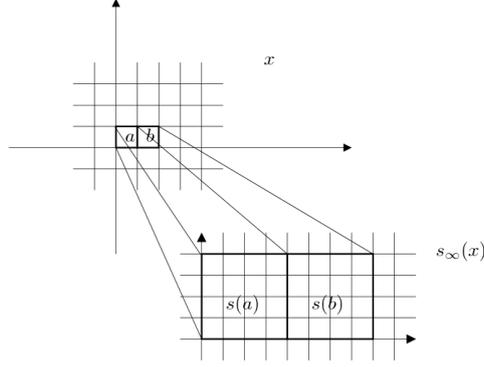
On appelle substitution s un ensemble fini de règles de substitution de la forme :

$$a \mapsto \begin{matrix} x_{k,1} & \dots & x_{k,k} \\ \vdots & & \vdots \\ x_{1,1} & \dots & x_{1,k} \end{matrix} \quad a, x_{i,j} \in \mathcal{A} ,$$

chaque règle étant unique pour une lettre de l'alphabet donnée. C'est donc une fonction $s : \mathcal{A} \rightarrow \mathcal{A}^{F_k}$ où $F_k = [1;k] \times [1;k]$. On peut étendre naturellement s en une fonction $s^n : \mathcal{A}^{F_n} \rightarrow \mathcal{A}^{F_{n \cdot k}}$. En partant d'une lettre située en $(1,1) \in \mathbb{Z}^2$ et en appliquant successivement $s, s^k, \dots, s^{k^{n-1}}$ on obtient une suite de motifs dans $\mathcal{A}^{F_{k^i}}$ pour $i = 0 \dots n$. De tels motifs sont appelés des s -motifs. Le sous-shift Σ_s défini par la substitution s est

$$\Sigma_s = \{x \in \mathcal{A}^{\mathbb{Z}^2} / \text{tout motif fini de } x \text{ apparaît dans un } s\text{-motif}\}$$

On définit $s_\infty : \Sigma_s \rightarrow \Sigma_s$ de la façon suivante : $s_\infty(x)$ est obtenu en appliquant la substitution s à chaque lettre de x : $(s_\infty(x))_u = (s(x_{u'}))_{u''}$ où $u' \in \mathbb{Z}^2$ et $u'' \in F_k$ sont tels que $u = ku' + u''$ (u' et u'' sont uniques).



Proposition 3 [Moz89] $x \in \Sigma_s$ si et seulement s'il existe $y \in \Sigma_s$ tel que $x = s_\infty(y)$. Si un tel y est unique pour tout x , on dit que la substitution s est à dérivation unique.

Un exemple de substitution, utilisé par Hochman dans [HM07], qui servira dans la suite est le suivant. On se place sur l'alphabet $\{\circ, \bullet\}$. Pour tout entier n la substitution s_n est donnée par :

$$\circ \mapsto \begin{matrix} \circ & \dots & \circ & \circ \\ \vdots & \cdot & \bullet & \circ \\ \circ & \cdot & \cdot & \vdots \\ \bullet & \circ & \dots & \circ \end{matrix} \quad \text{et} \quad \bullet \mapsto \begin{matrix} \circ & \dots & \circ & \bullet \\ \vdots & \cdot & \bullet & \circ \\ \circ & \cdot & \cdot & \vdots \\ \bullet & \circ & \dots & \circ \end{matrix}$$

où les motifs sont de taille $n * n$. On appelle Σ_n le sous-shift défini par la substitution s_n .

Proposition 4 Pour tout entier n , il existe un SFT $\tilde{\Sigma}_n$ et un morphisme lettre à lettre Π_n tel que $\Sigma_n = \Pi_n(\tilde{\Sigma}_n)$.

Ce résultat est en fait la conséquence d'un théorème beaucoup plus général dû à Mozes (théorème 4.5 de [Moz89]) : la conclusion reste vraie pour toute substitution à dérivation unique.

1.4.2 Exemple de la substitution s_p

On étudie plus en détails la substitution s_2 .

$$\circ \mapsto \begin{matrix} \circ & \circ \\ \bullet & \circ \end{matrix} \quad \text{et} \quad \bullet \mapsto \begin{matrix} \circ & \bullet \\ \bullet & \circ \end{matrix}$$

La simplicité de ce système (alphabet avec seulement deux éléments) permet de le décrire totalement : si $w \in \Sigma_2$ alors on sait exactement comment sont distribués les \circ et les \bullet .

Proposition 5 Si $w \in \Sigma_2$, on note $E(w) = \{u \in \mathbb{Z}^2 / w_u = \bullet\}$. Alors à une translation près on a :

$$E(w) = E = \bigcup_{n=1}^{\infty} (2^n \mathbb{Z} + 2^{n-1}) \times (2^n \mathbb{Z} + 2^{n-1})$$

Démonstration On construit la suite définie par $a_0 = \bullet$ et $a_1, a_2, \dots, a_n, \dots$ est la suite de motifs obtenue en appliquant la substitution s_2 à a_0 .

On commence par calculer le nombre de \bullet et de \circ dans a_n ; on note ces deux quantités b_n et c_n respectivement. Comme chaque \bullet de a_n donne deux \bullet dans a_{n+1} et que chaque \circ de a_n donne un \bullet dans a_{n+1} , on a les relations suivantes :

- $b_n + c_n = 4^n$
- $b_{n+1} = 2b_n + c_n$

desquelles on déduit $b_n = \sum_{k=0}^{n-1} 4^k$.

Revenons à la démonstration du résultat. Il suffit de montrer que $\{u \in F_{2^n} / (a_n)_u = \bullet\} = E \cap F_{2^n}$, puisque par définition du sous-shift tout motif autorisé apparaît dans un s -motif. On raisonne par récurrence sur n :

- pour $n = 1$ c'est évident d'après la règle de substitution ;
- supposons que $\{u \in F_{2^n} / (a_n)_u = \bullet\} = E \cap F_{2^n}$. Ou seront placés les \bullet dans a_{n+1} ? Pour un \bullet donné on a deux cas possibles :
 - soit il provient de la dérivation d'un \bullet en position $(2^k a + 2^{k-1}, 2^k b + 2^{k-1})$ avec $0 \leq a, b \leq 2^{n-k} - 1$ et $0 \leq k \leq n - 1$ dans a_n ; il est alors situé en $(2 \cdot (2^k a + 2^{k-1}) + 1, 2 \cdot (2^k b + 2^{k-1}) + 1) \in ((2\mathbb{Z} + 1) \times (2\mathbb{Z} + 1)) \cap F_{2^{n+1}}$ ou en $(2 \cdot (2^k a + 2^{k-1}) + 2, 2 \cdot (2^k b + 2^{k-1}) + 2) = (4 \cdot (2^{k-1} a + 2^{k-2}) + 2, 4 \cdot (2^{k-1} b + 2^{k-2}) + 2) \in ((4\mathbb{Z} + 2) \times (4\mathbb{Z} + 2)) \cap F_{2^{n+1}}$.
 - soit il provient de la dérivation d'un \circ en position (a, b) dans a_n ; il est situé en $(2a + 1, 2b + 1) \in ((2\mathbb{Z} + 1) \times (2\mathbb{Z} + 1)) \cap F_{2^{n+1}}$.

Dans tous les cas un \bullet de a_{n+1} est situé sur le réseau $E \cap F_{2^{n+1}}$. A nouveau un raisonnement par récurrence permet de montrer que $|E \cap F_{2^{n+1}}| = \sum_{k=0}^n 4^k = b_{n+1}$. Ainsi les \bullet qui apparaissent dans

a_{n+1} sont situés exactement sur ce réseau, ce qui achève la récurrence. \square

De la même façon on peut montrer que pour la substitution s_p avec p premier, l'ensemble E défini de manière analogue s'écrit à une translation près :

$$E = \bigcup_{n=1}^{\infty} \bigcup_{k=1}^{p-1} (p^n \mathbb{Z} + kp^{n-1}) \times (p^n \mathbb{Z} + kp^{n-1})$$

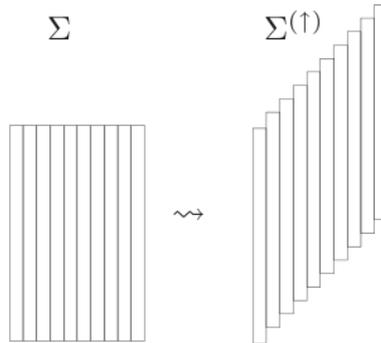
1.4.3 Construction d'espaces de calcul

Dans cette partie on explique comment utiliser les substitutions s_p pour construire un pavage non-périodique qui permettra de résoudre le problème des espaces de calcul pour machines de Turing. Pour cette construction on utilisera les sous-shifts Σ_2 et Σ_3 .

Si $\Sigma \subseteq \mathcal{A}^{\mathbb{Z}^2}$ est un sous-shift, on appelle $\Sigma^{(\uparrow)}$ le sous-shift défini par :

$$\Sigma^{(\uparrow)} = \{x \in \mathcal{A}^{\mathbb{Z}^2} / \exists y \in \Sigma, \forall (i, j) \in \mathbb{Z}^2, x_{(i, j)} = y_{(i, j-i)}\}$$

Cela revient à décaler toutes les configurations du sous-shift Σ de la manière suivante :

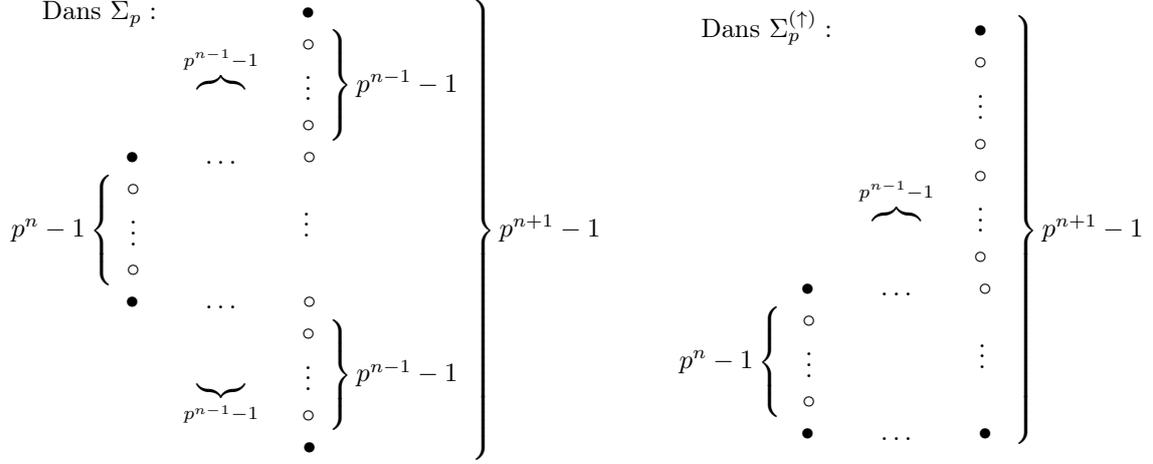


On notera que si Σ est un SFT, alors $\Sigma^{(\uparrow)}$ est aussi un SFT (il suffit de décaler les motifs interdits de Σ pour obtenir ceux de $\Sigma^{(\uparrow)}$).

Une conséquence de la structure de l'ensemble E est la suivante : on peut toujours trouver, sur une même ligne ou une même colonne dans Σ_p où les symboles \bullet apparaissent de façon p^n -périodique pour tout entier n .

Autre conséquence : si sur une même ligne (ou une même colonne) on a deux symboles \bullet à distance p^n , alors sur une autre ligne (ou une autre colonne) on a deux symboles \bullet à distance p^{n+1} . Si maintenant

on regarde $\Sigma_p^{(\uparrow)}$, les décalages synchronisent les colonnes de façon à ce que les \bullet s'alignent comme sur le dessin :



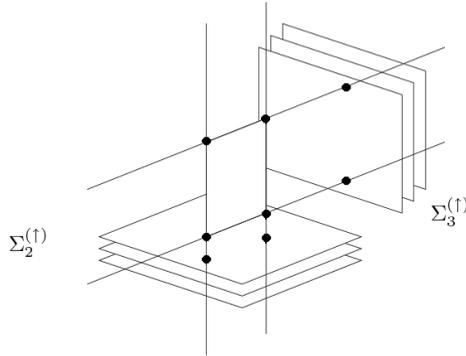
On se place dans l'espace $\mathbb{Z}^3 = \mathbb{Z}e_1 \oplus \mathbb{Z}e_2 \oplus \mathbb{Z}e_3$ et on construit les SFT W_2 et $W_3 \subseteq \{\circ, \bullet\}^{\mathbb{Z}^3}$ définis par :

$$x \in W_2 \Leftrightarrow \begin{cases} \forall k \in \mathbb{Z}, x|_{\mathbb{Z}^2 \times \{k\}} \in \Sigma_2^{(\uparrow)} \\ \forall u \in \mathbb{Z}^3, x_u = x_{u+e_3} \quad (*) \end{cases}$$

$$x \in W_3 \Leftrightarrow \begin{cases} \forall j \in \mathbb{Z}, x|_{\mathbb{Z} \times \{j\} \times \mathbb{Z}} \in \Sigma_3^{(\uparrow)} \\ \forall u \in \mathbb{Z}^3, x_u = x_{u+e_2} \quad (**) \end{cases}$$

On appellera **rectangle** toute partie de \mathbb{Z}^3 de la forme $\{i\} * [j; j+l] * [k; k+h]$; l la largeur du rectangle et h sa hauteur et on dira que le rectangle est situé en (i, j, k) .

Propriété 1 *Le produit $W_2 \times W_3$ dessine un découpage de l'espace en rectangles, de sorte que chaque plan $\{i\} \times \mathbb{Z}^2$ est pavé par des rectangles de même taille. De plus si on a un rectangle de taille $2^n * 3^m$ en $(i, j, k) \in \mathbb{Z}^3$, alors il existe i' et i'' tel qu'on ait un rectangle de taille $2^{n+1} * 3^m$ en (i', j, k) et un rectangle de taille $2^n * 3^{m+1}$ en (i'', j, k)*



Démonstration On se place sur un plan $\{i\} \times \mathbb{Z}^2$. Alors il lui correspond une colonne dans $\Sigma_2^{(\uparrow)}$ et une colonne dans $\Sigma_3^{(\uparrow)}$. Sur ces colonnes les \bullet sont placés de manière 2^n -périodique et 3^m -périodique respectivement. Les lettres (\bullet, \bullet) de l'alphabet produit forment donc des rectangles de taille $2^n * 3^m$ grâce aux règles $(*)$ et $(**)$.

Regardons à présent un tel rectangle, de taille $2^n * 3^m$ situé en (i, j, k) . Alors pour tout $i_1 = i + 2^{n-1} + \lambda 2^{n+1}$ on a des rectangles de largeur 2^{n+1} . De même pour tout $i_2 = i + \mu 3^m$ on a des rectangles de hauteur 3^m . On cherche donc des entiers λ et μ tels que

$$i + \mu 3^m = i + 2^{n-1} + \lambda 2^{n+1} \Leftrightarrow \mu 3^m = 2^{n-1}(1 + 4\lambda)$$

ce qui est possible car 3^m et 2^n sont premiers entre eux. On peut donc trouver i' tel qu'on ait un rectangle de taille $2^{n+1} * 3^m$ en (i', j, k) . Le raisonnement est identique pour i'' . \square

2 Sous-shifts, sous-actions et facteurs

On s'intéresse à deux opérations sur les sous-shifts : facteur et sous-action. Si $\Sigma \subset \mathcal{A}^{\mathbb{Z}^d}$ est un sous-shift, on essaie de le transformer en un autre sous-shift soit par des modifications locales l'alphabet de départ \mathcal{A} (prendre un facteur de Σ) soit en regardant un sous-système de Σ (prendre une sous-action de Σ).

La classe des SFT n'étant stable pour aucune de ces deux actions, on va chercher des classes de sous-shifts contenant les SFT qui soit stable par facteur d'une part et par sous-action d'autre part.

2.1 Facteur d'un sous-shift

Soient \mathcal{A} et \mathcal{B} deux alphabets finis. On se donne $\Sigma \subset \mathcal{A}^{\mathbb{Z}^d}$ un sous-shift.

2.1.1 Définition et exemples

Un **morphisme** $\Pi : \mathcal{A}^{\mathbb{Z}^d} \rightarrow \mathcal{B}^{\mathbb{Z}^d}$ est une application continue et qui commute avec le décalage $\sigma : \forall u \in \mathbb{Z}^d : \sigma^u \circ \Pi = \Pi \circ \sigma^u$. D'après [Hed69] cela revient à dire qu'il existe $\mathbb{V} \subset \mathbb{Z}^d$ un voisinage fini et $\pi : \mathcal{A}^{\mathbb{V}} \rightarrow \mathcal{B}$ tels que $\forall x \in \mathcal{A}^{\mathbb{Z}^d}, \forall u \in \mathbb{Z}^d, (\Pi(x))_u = \pi(x_{u+\mathbb{V}})$. Si $\Sigma \subseteq \mathcal{A}^{\mathbb{Z}^d}$ et $\Sigma' \subseteq \mathcal{B}^{\mathbb{Z}^d}$ sont deux sous-shifts, on dit que Σ' est un **facteur** de Σ s'il existe un morphisme Π tel que $\Sigma' = \Pi(\Sigma)$.

2.1.2 Sous-shift sofique

Qu'obtient-on en appliquant un facteur à un SFT ? Comme le montre l'exemple suivant un facteur de SFT n'est pas un SFT :

Exemple 1 *On se place dans $\{0, 1, 2\}^{\mathbb{Z}}$ et on définit $\Sigma = X_{\{00, 11, 02, 21\}}$. On définit le facteur Π par $\Pi(0) = \Pi(1) = 0$ et $\Pi(2) = 2$. Alors :*

$$\Pi(\Sigma) = \{x \in \{0, 2\}^{\mathbb{Z}} / \text{les blocs de } 0 \text{ consécutifs sont de longueurs paires}\}$$

qui n'est pas un SFT, puisque pour le décrire il faut exclure les blocs de 0 de taille impaires de longueur arbitrairement grande.

En appliquant des facteurs aux SFT, on obtient une nouvelle classe de sous-shifts : les sous-shifts soifiques. Comme la composée de deux facteurs reste un facteur, c'est la plus petite classe de sous-shifts contenant les SFT et stable par facteur. En dimension 1, on connaît une caractérisation des soifiques qui permet de faire le lien avec la théorie des langages et d'obtenir toute une série d'exemples de tels sous-shifts.

Théorème 1 [LM95] *En dimension 1, les sous-shifts soifiques sont exactement les sous-shifts donnés par un ensemble de motifs interdits rationnel.*

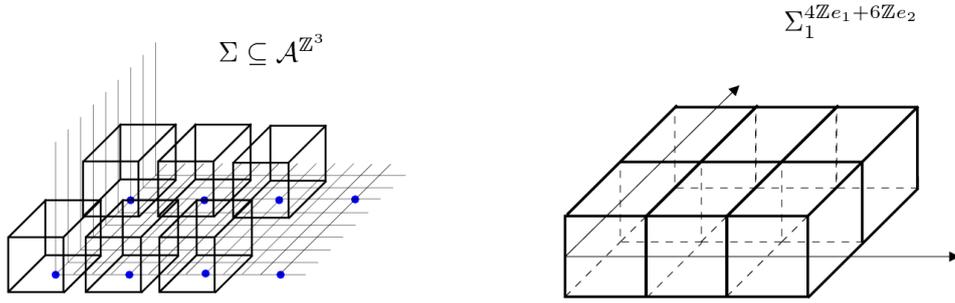
2.2 Sous-action

Si Σ est un sous-shift de dimension d , on peut se demander ce qu'on obtient en restreignant Σ à un réseau de dimension inférieure à d .

2.2.1 Définition

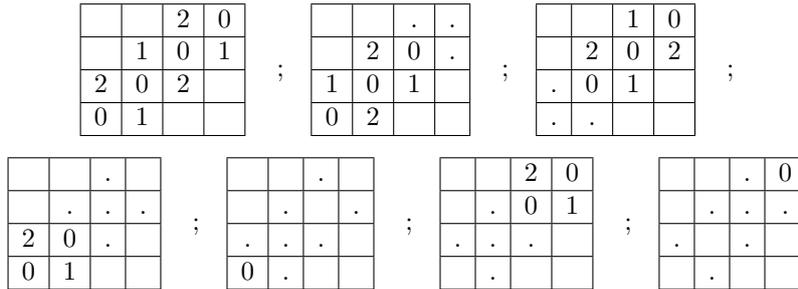
Soit $\phi_k : \mathcal{A}^{\mathbb{Z}^d} \rightarrow \mathcal{A}^{[-k;k]^d}$ l'application qui à une configuration du réseau associe le cube de côté $2k + 1$ centré en l'origine. Si G est un sous-groupe de \mathbb{Z}^d engendré par les vecteurs $u_1, \dots, u_{d'}$ $\in \mathbb{Z}^d$, on note $\Phi_k(x) = y$ si $\forall (i_1, \dots, i_{d'}) \in \mathbb{Z}^{d'}, y_{i_1, \dots, i_{d'}} = \phi_k(\sigma^{i_1 * u_1 + \dots + i_{d'} * u_{d'}}(x))$ et on définit $\Sigma_k^G = \{y \in (\mathcal{A}^{[-k;k]^d})^{\mathbb{Z}^{d'}} / \exists x \in \Sigma, \Phi_k(x) = y\}$. C'est l'image du sous-shift Σ par la projection sur le sous-groupe G et d'épaisseur k .

Si $\Sigma \subseteq \mathcal{A}^{\mathbb{Z}^d}$ est un sous-shift, une sous-action de Σ est de la forme Σ_k^G avec $k \in \mathbb{N}^*$ et $G < \mathbb{Z}^d$.



Propriété 2 La classe des SFT n'est pas stable par sous-action.

Démonstration On construit $\Sigma \subset \{0, 1, 2\}^{\mathbb{Z}^2}$ un SFT tel que la projection de Σ sur la droite Δ d'équation $y = x$ n'est pas de type fini. Dans cet exemple on veut que le sous-shift qui apparaît sur la droite soit $\{x \in \{0, 1, 2\}^{\mathbb{Z}} / \text{les blocs de 0 consécutifs sont de longueurs paires}\}$, qui n'est pas un SFT comme on l'a vu précédemment. On décrit un ensemble de motifs autorisés \overline{F} (le symbole \cdot peut être 1 ou 2 mais pas 0, le symbole blanc peut être 0,1 ou 2) :



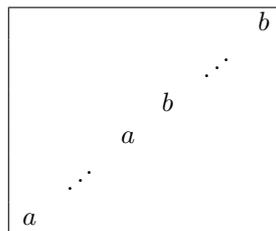
C'est l'alternance de 1 et de 2 sur et sous la diagonale de 0 qui permet de maîtriser la parité des longueurs de blocs de 0. On définit F comme l'ensemble des motifs élémentaires de côté 4 qui ne sont pas dans \overline{F} . Alors en notant $\Sigma = X_F$:

$$\Sigma_0^\Delta = \{x \in \{0, 1, 2\}^{\mathbb{Z}} / \text{les blocs de 0 consécutifs sont de longueurs paires}\}$$

□

Propriété 3 La classe des sofiques n'est pas stable par sous-action.

Démonstration On construit un décalage sofique Σ tel que la projection Σ_0^Δ sur la droite d'équation $y = x$ n'est pas sofique. On souhaite faire apparaître sur cette diagonale un langage non rationnel, par exemple $\{a^n b^n / n \in \mathbb{N}\}$. On cherche donc Σ' un SFT et Π un morphisme tels que les seules tuiles autorisées dans $\Sigma = \Pi(\Sigma')$ contenant a ou b sont celles de la forme $2n * 2n$:



Comment construire cet ensemble de motifs? On veut des motifs carrés de côté pair, et savoir positionner le centre du carré (pour distinguer la partie a^n de la partie b^n).

On appelle $\mathcal{A} = \{*, \#, 0, 1, 2, 3, 4\}$. Les symboles $*$ et $\#$ serviront à dessiner les diagonales des tuiles; les 1, 2, 3, 4 à distinguer dans quelle partie de la tuile on se trouve. On construit un jeu de tuiles qui n'autorise la construction que de carrés de côté pair de la forme :

$$\begin{array}{cccccc} * & 1 & \dots & \dots & 1 & \# \\ 4 & \cdot & 1 & 1 & \cdot & 2 \\ \vdots & 4 & * & \# & 2 & \vdots \\ \vdots & 4 & \# & * & 2 & \vdots \\ 4 & \cdot & 3 & 3 & \cdot & 2 \\ \# & 3 & \dots & \dots & 3 & * \end{array} \quad (*)$$

Voici l'ensemble des motifs autorisés :

Centre des carrés :

$$\begin{array}{cccc} * & 1 & 1 & \# \\ 4 & * & \# & 2 \\ 4 & \# & * & 2 \\ \# & 3 & 3 & * \end{array}$$

Diagonales des carrés :

$$\begin{array}{ccc} \begin{array}{ccc} * & 1 & 1 \\ 4 & * & 1 \\ 4 & 4 & * \end{array} & \begin{array}{ccc} 1 & 1 & \# \\ 1 & \# & 2 \\ \# & 2 & 2 \end{array} & \begin{array}{ccc} * & 2 & 2 \\ 3 & * & 2 \\ 3 & 3 & * \end{array} & \begin{array}{ccc} 4 & 4 & \# \\ 4 & \# & 3 \\ \# & 3 & 3 \end{array} \end{array}$$

Bords des carrés :

$$\begin{array}{ccccc} \begin{array}{ccc} 0 & 0 & 0 \\ 0 & * & 1 \\ 0 & 4 & * \end{array} & \begin{array}{ccc} 0 & 0 & 0 \\ 1 & 1 & 1 \\ * & 1 & 1 \end{array} & \begin{array}{ccc} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array} & \begin{array}{ccc} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & \# \end{array} & \begin{array}{ccc} 0 & 0 & 0 \\ 1 & \# & 0 \\ \# & 2 & 0 \end{array} \end{array}$$

... même principe pour les trois autres côtés.

Remplissage des carrés :

$$\begin{array}{cccc} \begin{array}{cc} 1 & 1 \\ 1 & 1 \end{array} & \begin{array}{cc} 2 & 2 \\ 2 & 2 \end{array} & \begin{array}{cc} 3 & 3 \\ 3 & 3 \end{array} & \begin{array}{cc} 4 & 4 \\ 4 & 4 \end{array} \end{array}$$

Extérieur des carrés : $\boxed{0}$

Avec ces seuls motifs autorisés, on peut construire des configurations de $\mathcal{A}^{\mathbb{Z}^2}$ avec des 0 partout sauf en certains endroits où l'on a des blocs de la forme (*) de tailles arbitraires (et aussi la configuration faite du motif (*) étendu à l'infini). On note Σ' ce SFT.

Il reste à construire un morphisme Π convenable, $\Pi : (\mathcal{A})^{\mathbb{Z}^2} \rightarrow \{0, a, b\}^{\mathbb{Z}^2}$. On décrit Π par des règles locales (le symbole \cdot peut être remplacé par n'importe quelle lettre de \mathcal{A}) :

Première moitié de la diagonale :

$$\begin{array}{ccc} \begin{array}{ccc} \cdot & \cdot & \cdot \\ \cdot & \# & 3 \\ \cdot & \cdot & \cdot \end{array} & \text{ou} & \begin{array}{ccc} \cdot & \cdot & \cdot \\ \cdot & \# & \cdot \\ \cdot & 3 & \cdot \end{array} \rightarrow a \end{array}$$

Seconde moitié de la diagonale :

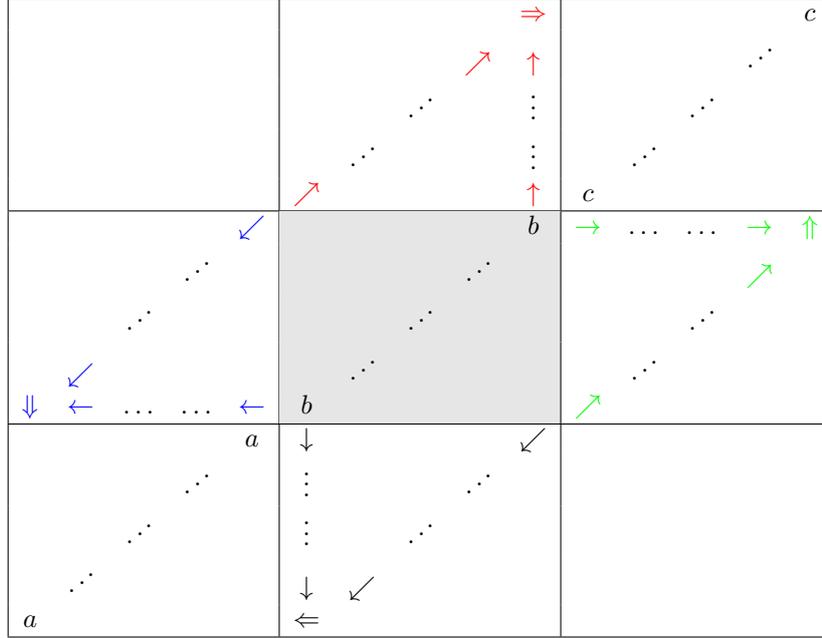
$$\begin{array}{ccc} \begin{array}{ccc} \cdot & \cdot & \cdot \\ \cdot & \# & 2 \\ \cdot & \cdot & \cdot \end{array} & \text{ou} & \begin{array}{ccc} \cdot & \cdot & \cdot \\ \cdot & \# & \cdot \\ \cdot & 2 & \cdot \end{array} \rightarrow b \end{array}$$

Π étant défini par des règles locales il est continu et commute avec σ le décalage, c'est bien un morphisme. On note $\Sigma = \Pi(\Sigma')$ qui est donc un décalage sofique. Ainsi $\Sigma_0^\Delta = \{x \in \{0, a, b\}^{\mathbb{Z}^2} / \text{tous les blocs formés uniquement de } a \text{ et de } b \text{ sont de la forme } a^n b^n\}$. Comme le langage $\{a^n b^n, n \in \mathbb{N}\}$ n'est pas un langage rationnel, Σ_0^Δ n'est pas sofique. \square

Ce contre-exemple fonctionne car on fait apparaître sur la diagonale un langage qui n'est pas reconnu par automate fini. Cependant il est reconnu par automate à pile. On peut construire d'autres contre-exemples plus pertinent dans lesquels le langage obtenu sur la diagonale n'est pas reconnaissable par automate à pile, par exemple :

$$\mathcal{L} = \{x \in (\{0, a, b, c\})^{\mathbb{Z}} / \text{tout bloc formé de } a, b \text{ et } c \text{ est de la forme } a^n b^n c^n, n \in \mathbb{N}\}$$

On commence par former le carré central (zone grisée dans la figure) qui contiendra la partie b^n du motif $a^n b^n c^n$.



On fait ensuite partir des signaux des quatre coins de ce carré comme expliqué sur le dessin, de sorte que les carrés au-dessus et en dessous du précédent, contenant les c^n et a^n respectivement, auront la taille voulue.

2.2.2 Sous-shifts récursivement énumérables

Dans cette partie on s'intéresse aux sous-shifts récursivement énumérables qui sont les sous-shifts dont l'ensemble des motifs interdits est récursivement énumérable. On notera simplement sous-shift RE dans la suite. Ces sous-shifts ont une complexité suffisante pour être stable par sous-action, comme le montre le résultat suivant.

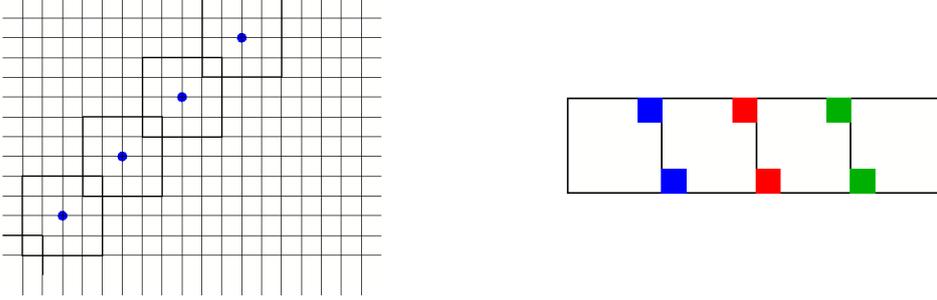
Théorème 2 *La classe des sous-shifts récursivement énumérables est stable par sous-action.*

Démonstration du théorème Soit $\Sigma \subseteq \mathcal{A}^{\mathbb{Z}^d}$ un sous-shift récursivement énumérable donné par F un ensemble récursivement énumérable de motifs interdits. Soit $\mathbb{G} < \mathbb{Z}^d$ un sous-groupe engendré par les vecteurs $u_1, \dots, u_{d'} \in \mathbb{Z}^d$, et k un entier non nul. Le principe de la démonstration est le suivant : à partir de l'ensemble F on va construire un ensemble F' lui aussi RE tel que $\Sigma_k^{\mathbb{G}} = X_{F'}$. L'ensemble F' sera l'ensemble de tous les motifs interdits, c'est-à-dire le complémentaire du langage de $\Sigma_k^{\mathbb{G}}$.

On note \mathcal{B} l'alphabet $\mathcal{A}^{[-k;k]^d}$, de sorte que $\Sigma_k^{\mathbb{G}} \subseteq \mathcal{B}^{\mathbb{Z}^{d'}}$. Pour montrer le théorème il suffit de construire un ensemble F' possédant les propriétés suivantes : F' est RE et $\Sigma_k^{\mathbb{G}} = X_{F'}$. Si m' est un motif de $\mathcal{B}^{[-n;n]^{d'}}$, on peut voir m' comme un motif de \mathcal{A}^I avec $I = \cup_{i=(i_1, \dots, i_{d'}) \in [-n;n]^{d'}} (i_1 * u_1 + \dots + i_{d'} * u_{d'}) + [-k;k]^d$ et on note $\Phi_k^{-1}(m')$ est l'ensemble des manières de compléter ce motif en un motif élémentaire de taille quelconque.

On définit F' de la façon suivante : un motif $m' \in \mathcal{B}^{[-n;n]^{d'}}$ est dans F' si et seulement si :

- il existe $n \in \mathbb{N}$ tel que tout motif de $\Phi_k^{-1}(m') \cap \mathbb{S}_n^d$ contient un motif de F ;
- m' respecte d'éventuels chevauchements si k est grand devant les $\|u_i\|$ (voir la figure).



Propriété 4 *L'ensemble F' est RE.*

Démonstration Il est facile de construire une machine de Turing \mathcal{M}_{chev} qui décide si un motif $m' \in \mathcal{B}^{[-n;n]^{d'}}$ respecte les chevauchements.

Comme F est RE, il existe une machine de Turing \mathcal{M} qui accepte tout motif de F et boucle sur les autres entrées. Pour tout entier $p \in \mathbb{N}$ on construit une machine de Turing $\mathcal{M}_{int}^{(p)}$ avec le comportement suivant : on lui donne en entrée un motif $m' \in \mathcal{B}^{[-n;n]^{d'}}$; la machine $\mathcal{M}_{int}^{(p)}$:

- accepte m' si quelque soit la façon de compléter m' en un motif $m \in \Phi_k^{-1}(m') \cap \mathcal{A}^{\mathbb{S}_p^d}$, m contient un mot interdit de Σ ,
- boucle sinon.

Décrivons le fonctionnement de la machine $\mathcal{M}_{int}^{(p)}$. On énumère toutes les façons de compléter m' en un motif $m \in \Phi_k^{-1}(m') \cap \mathcal{A}^{\mathbb{S}_p^d}$. Il y en a un nombre fini, que l'on note m_1, \dots, m_N . Pour chaque motif m_i complétant m' de cette manière, on simule le comportement de \mathcal{M} sur chacun des sous-motifs élémentaires de m_i de la manière suivante :

- On suppose que l'on énumère tous les sous-motifs élémentaires de m_i (par exemple par taille croissante, puis par coordonnées croissantes) : on les appelle $m_i^1, \dots, m_i^{N_i}$.
- On simule la première étape du calcul de \mathcal{M} sur m_1
- On simule la première étape du calcul de \mathcal{M} sur m_2
- On simule la deuxième étape du calcul de \mathcal{M} sur m_1
- On simule la deuxième étape du calcul de \mathcal{M} sur m_2
- On simule la première étape du calcul de \mathcal{M} sur m_3
- On simule la troisième étape du calcul de \mathcal{M} sur m_1
- On simule la troisième étape du calcul de \mathcal{M} sur m_2
- On simule la deuxième étape du calcul de \mathcal{M} sur m_3
- On simule la première étape du calcul de \mathcal{M} sur m_4
- ...
- Dès qu'un des calculs de \mathcal{M} sur un des m_i termine, on arrête le processus. Sinon on boucle.

On appelle cette série d'étapes le calcul de $\mathcal{M}_{int}^{(p)}$ sur le motif m_i . La machine $\mathcal{M}_{int}^{(p)}$ procède de la manière suivante :

- On simule la première étape du calcul de $\mathcal{M}_{int}^{(p)}$ sur m_1
- On simule la première étape du calcul de $\mathcal{M}_{int}^{(p)}$ sur m_2
- On simule la deuxième étape du calcul de $\mathcal{M}_{int}^{(p)}$ sur m_1
- On simule la deuxième étape du calcul de $\mathcal{M}_{int}^{(p)}$ sur m_2
- On simule la première étape du calcul de $\mathcal{M}_{int}^{(p)}$ sur m_3
- ...

Si la machine $\mathcal{M}_{int}^{(p)}$ accepte m' , alors tout motif $m \in \mathcal{A}^{\mathbb{S}_p^d}$ complétant m' contient un motif interdit dans Σ donc m' est dans $(\mathcal{L}(\Sigma_k^G))^c$. Mais si $\mathcal{M}_{int}^{(p)}$ boucle sur m' cela ne veut pas dire pour autant que

m' est dans $\mathcal{L}(\Sigma_k^{\mathbb{G}})$. Pour avoir $m' \in \mathcal{L}(\Sigma_k^{\mathbb{G}})$ il faut que toutes les machines $\mathcal{M}_{int}^{(p)}$ bouclent sur m' . La machine \mathcal{M}_{int} aura donc le comportement suivant sur une entrée m' :

- \mathcal{M}_{int} calcule le plus petit entier p tel que $\Phi_k^{-1}(m') \cap \mathcal{A}^{\mathbb{S}_p^d} \neq \emptyset$
- on simule la première étape du calcul de $\mathcal{M}_{int}^{(p)}$ sur m'
- on simule la première étape du calcul de $\mathcal{M}_{int}^{(p+1)}$ sur m'
- on simule la deuxième étape du calcul de $\mathcal{M}_{int}^{(p)}$ sur m'
- on simule la deuxième étape du calcul de $\mathcal{M}_{int}^{(p+1)}$ sur m'
- on simule la première étape du calcul de $\mathcal{M}_{int}^{(p+2)}$ sur m'
- ...

La machine \mathcal{M}_{int} s'arrête dès qu'un des calcul des $\mathcal{M}_{int}^{(p)}$ s'arrête. Ainsi \mathcal{M}_{int} s'arrête sur m' si et seulement si il existe un entier p tel que tous les motifs $m \in \Phi_k^{-1}(m') \cap \mathcal{A}^{\mathbb{S}_p^d}$ contiennent un motif interdit.

Pour vérifier qu'un motif m' est dans F' , il suffit de simuler le comportement de la machine \mathcal{M}_{int} sur m' , puis si le motif est accepté par \mathcal{M}_{int} de simuler \mathcal{M}_{chev} sur m' . On a ainsi une machine de Turing qui accepte m' si $m' \in F'$ et boucle sinon. L'ensemble F' est donc récursivement énumérable. \square

Propriété 5 $\Sigma_k^{\mathbb{G}} = X_{F'}$.

Démonstration

- $\Sigma_k^{\mathbb{G}} \subseteq X_{F'}$
Soit $y \in \Sigma_k^{\mathbb{G}}$. Alors il existe un $x \in \Sigma$ tel que pour tout $i = (i_1, \dots, i_{d'}) \in \mathbb{Z}^{d'}$, $y_i = \phi_k(\sigma^{i_1 * u_1 + \dots + i_{d'} * u_{d'}}(x))$. Soit m' un motif de y . Si par l'absurde m' était dans F' tout motif complétant m' contiendrait un motif interdit dans Σ , donc en particulier x contiendrait un motif interdit dans Σ ce qui voudrait dire que $x \notin \Sigma$: contradiction.
 y ne contient donc aucun motif de F' , ce qui montre que $y \in X_{F'}$.
- $X_{F'} \subseteq \Sigma_k^{\mathbb{G}}$
Soit $y \in X_{F'}$. On a donc $y = (y_i)_{i \in \mathbb{Z}^{d'}}$ avec pour tous $i \in \mathbb{Z}^{d'}$ et $n \in \mathbb{N}$, $y_{i+[-n;n]^{d'}} \notin F'$. Pour tout $n \in \mathbb{N}$, on a $y_{[-n;n]^{d'}} \notin F'$ ce qui signifie qu'il existe un motif $m_n \in \Phi_k^{-1}(y_{[-n;n]^{d'}})$ complétant $y_{[-n;n]^{d'}}$ et qui ne contient aucun motif interdit dans Σ . m_n est un motif fini, on le complète de façon à avoir un $x_n \in (\mathcal{A} \cup \{\#\})^{\mathbb{Z}^d}$, où $\#$ est un symbole qui n'est pas dans \mathcal{A} . Ainsi $x_n \in \Sigma_{\#}$ où $\Sigma_{\#} \subseteq (\mathcal{A} \cup \{\#\})^{\mathbb{Z}^d}$ est le sous-shift défini avec le même ensemble de mots interdits que Σ . On a une suite $(x_n)_{n \in \mathbb{N}}$ dans l'espace $(\mathcal{A} \cup \{\#\})^{\mathbb{Z}^d}$ qui est compact en tant qu'espace produit, donc on peut extraire de $(x_n)_{n \in \mathbb{N}}$ une suite $(x_{\phi(n)})_{n \in \mathbb{N}}$ qui converge vers $x \in (\mathcal{A} \cup \{\#\})^{\mathbb{Z}^d}$. Par construction des x_n on a même que $x \in \mathcal{A}^{\mathbb{Z}^d}$ et comme $\Sigma_{\#}$ est fermé, on a $x \in \Sigma_{\#}$, d'où $x \in \Sigma_{\#} \cap \mathcal{A}^{\mathbb{Z}^d} = \Sigma$. Comme x complète y , on a bien $y_i = \Phi_k(\sigma^{i_1 * u_1 + \dots + i_{d'} * u_{d'}}(x))$ pour tout $i = (i_1, \dots, i_{d'}) \in \mathbb{Z}^{d'}$. Ceci montre $X_{F'} \subseteq \Sigma_k^{\mathbb{G}}$. \square

On a construit un ensemble récursivement énumérable F' tel que $X_{F'} = \Sigma_k^{\mathbb{G}}$, ce qui montre que la classe des sous-shifts récursivement énumérables est stable par sous-action. \square

Question : Que dire des sous-shifts algébriques et récursifs, qui seraient les sous-shifts dont l'ensemble des motifs interdits est soit algébrique (pour la dimension 1 uniquement) soit récursif (en dimension quelconque) ?

3 Clôture de classes des sous-shifts

Dans cette partie on décrit de nouvelles opérations sur les sous-shifts, et on s'intéresse au problème suivant : qu'obtient-on en appliquant certaines de ces opérations à une classe de sous-shifts ? A partir d'un ensemble d'opérations on définit un ordre \leq_{op} sur les sous-shifts. On cherche à faire correspondre cet ordre sur les sous-shifts avec un ordre sur les langages \preceq_{op} de façon à avoir :

$$X_{\mathcal{L}} \leq_{op} X_{\mathcal{L}'} \Leftrightarrow \mathcal{L} \preceq_{op} \mathcal{L}'$$

3.1 Opérations et clôtures

On note \mathcal{S} l'ensemble des sous-shifts (de toutes dimensions et sur tous les alphabets), \mathcal{S}^d l'ensemble des sous-shifts de dimension d et $\mathcal{S}_{\mathcal{A}}$ l'ensemble des sous-shifts d'alphabet \mathcal{A} .

Soit \mathcal{U} un ensemble de sous-shifts. Pour tout d , on note $\mathcal{U}^d = \mathcal{U} \cap \mathcal{S}^d$ et $\mathcal{U}_{\mathcal{A}} = \mathcal{U} \cap \mathcal{S}_{\mathcal{A}}$. On définit $Cl_i(\mathcal{U})$ la **clôture** de la classe \mathcal{U} par l'opération i comme l'ensemble de tous les sous-shifts obtenus en appliquant un nombre fini de fois l'opération i aux sous-shifts de \mathcal{U} . En particulier pour toute opération i on a $\mathcal{U} \subset Cl_i(\mathcal{U})$.

- **Produit (P)**

On fait le produit de deux sous-shifts de même dimension. Cela revient à superposer les configurations. Si $\Sigma = X_F \subseteq \mathcal{A}^{\mathbb{Z}^d}$ et $\Sigma' = X_{F'} \subseteq \mathcal{B}^{\mathbb{Z}^d}$ sont deux sous-shifts de dimension d , leur produit est : $\Sigma \times \Sigma' = X_{F \times F'} \subseteq (\mathcal{A} \times \mathcal{B})^{\mathbb{Z}^d}$. Cette opération permet de faire grossir l'alphabet d'un sous-shift. Par exemple $\Sigma \times \mathcal{B}^{\mathbb{Z}^d}$ est le sous-shift inclus dans $(\mathcal{A} \times \mathcal{B})^{\mathbb{Z}^d}$ défini par les mêmes motifs interdits que Σ .

- **Facteur (F)**

C'est l'opération de facteur déjà étudiée en 2.1.1. Si $\Sigma \subseteq \mathcal{A}^{\mathbb{Z}^d}$ est un sous-shift sur l'alphabet \mathcal{A} on le transforme par des modifications locales en sous-shift de même dimension sur un alphabet \mathcal{B} .

- **Injection (I)**

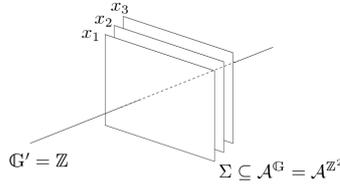
C'est en quelque sorte l'opération complémentaire du facteur. Si $\Sigma \subseteq \mathcal{A}^{\mathbb{Z}^d}$ et $\Pi : \mathcal{B}^{\mathbb{Z}^d} \rightarrow \mathcal{A}^{\mathbb{Z}^d}$ est un morphisme injectif, alors le sous-shift Σ' tel que $\Pi(\Sigma') = \Sigma$ est une injection de Σ .

- **Sous-action (SA)**

Cette opération a été définie en 2.2.1. Si $\Sigma \subseteq \mathcal{A}^{\mathbb{Z}^d}$ est un sous-shift de dimension d , on s'intéresse aux sous-shifts de dimension inférieure (ou égale) contenus dans Σ . Ce sont les $\Sigma_k^{\mathbb{G}}$ avec $k \in \mathbb{N}$ et $\mathbb{G} < \mathbb{Z}^d$.

- **Superposition (SP)**

On augmente la dimension d'un sous-shift en superposant des copies indépendantes. Si $\Sigma \subseteq \mathcal{A}^{\mathbb{Z}^d}$ est un sous-shift et $\mathbb{Z}^d \simeq \mathbb{G} < \mathbb{G} \oplus \mathbb{G}' \simeq \mathbb{Z}^{d+d'}$ alors la superposition de Σ dans $\mathbb{G} \oplus \mathbb{G}'$ selon la direction \mathbb{G}' est $\phi_{SP}(\Sigma, \mathbb{G}, \mathbb{G}') = \{x \in \mathcal{A}^{\mathbb{Z}^{d+d'}} / \forall u \in \mathbb{G}', x|_{\mathbb{G}+u} \in \Sigma\}$.



- **Type fini (TF)**

Cette opération consiste à rajouter des conditions de type fini à un sous-shift : si $\Sigma \subseteq \mathcal{A}^{\mathbb{Z}^d}$ et F_1, \dots, F_p sont des cylindres de Σ , on construit $\phi_{TF}(\Sigma, \{F_1, \dots, F_p\}) = \Sigma \cap \left(\bigcap_{i=1}^p \bigcap_{u \in \mathbb{Z}^d} \sigma^u(F_i) \right)$.

Si on choisit plusieurs opérations dans la liste ci-dessus, la clôture d'une classe \mathcal{U} est l'ensemble des sous-shifts obtenus en appliquant un nombre fini de ces opérations à un élément de \mathcal{U} . On peut ainsi traduire en terme de clôture certains résultats comme on le verra dans la partie suivante.

Si op est un ensemble d'opérations, on définit un pré-ordre (relation réflexive et transitive) sur les sous-shifts par :

$$\Sigma \leq_{op} \Sigma' \Leftrightarrow \Sigma \in Cl_{op}(\Sigma')$$

3.2 Traduction de résultats en terme de clôture

On traduit ici à l'aide de clôture des résultats de stabilité de classes de sous-shifts. On commence par définir des classes de sous-shift :

- \mathcal{SFT} est l'ensemble de tous les SFT (en toute dimension et sur tout alphabet)
- $\mathcal{Sofique}$ est l'ensemble de tous les sous-shifts sofiques
- \mathcal{RE} est l'ensemble des sous-shifts récursivement énumérables $\mathcal{RE} = \{X_F / F \text{ est RE} \}$

- \mathcal{FS} l'ensemble des full-shifts, qui sont de la forme $\mathcal{A}^{\mathbb{Z}^d}$

A l'aide de ces nouvelles notations on peut écrire de manière plus concise des résultats déjà connus, notamment ceux concernant la plus petite classe étudiée, celle des SFT :

- les SFT sont stables par produit et superposition : $Cl_P(\mathcal{SFT}) = \mathcal{SFT}$ et $Cl_{SP}(\mathcal{SFT}) = \mathcal{SFT}$
- un SFT est un full-shift auquel on a ajouté des conditions de type fini : $Cl_{TF}(\mathcal{FS}) = \mathcal{SFT}$
- les sofiques sont images d'un SFT par un facteur : $Cl_F(\mathcal{SFT}) = \mathcal{Sofique}$
- en dimension 1 on sait caractériser les sofiques par leur langage : $Cl_F(\mathcal{SFT}) \cap \mathcal{S}^1 = \{X_{\mathcal{L}}/\mathcal{L} \text{ est rationnel}\}$

Dans la partie 2 on a aussi montré que $Cl_F(\mathcal{Sofique}) = \mathcal{Sofique}$ et $Cl_{SA}(\mathcal{RE}) = \mathcal{RE}$. Ce sont des résultats de stabilité, et on aimerait obtenir des résultats de simulation : étant données une classe de sous-shifts et des opérations, quelle classe obtient-on par clôture ?

Dans [Hoc07b] on trouve un résultat de simulation : $Cl_{F,SA}(\mathcal{SFT}) = \{X_{\mathcal{L}}/\mathcal{L} \text{ est RE}\}$. C'est en fait plus précis qu'une simple clôture, puisque pour obtenir n'importe quel sous-shift RE de dimension d , on a besoin d'un SFT de dimension $d+2$ et ce pour d quelconque.

Question : Peut-on simuler un sous-shift RE de dimension d par un SFT de dimension $d+1$? Si on veut reprendre le même schéma de preuve, il faudrait arriver à construire des rectangles comme en 1.4.3 mais en utilisant seulement 2 dimensions au lieu de 3.

On s'intéresse maintenant à l'action de ces opérations non plus sur une classe de sous-shifts mais sur un seul sous-shift, et on cherche à caractériser la clôture de $X_{\mathcal{L}}$. [Hoc07a] montre l'existence d'un SFT de dimension 3 Σ tel que :

$$Cl_{F,SA}(\Sigma) = \{X_{\mathcal{L}} \in \mathcal{S}^1/\mathcal{L} \text{ est RE}\}$$

c'est-à-dire un SFT de dimension 3 universel pour la classe des sous-shifts RE de dimension 1, mais avec comme alphabet un ensemble de Cantor.

3.3 Un ordre sur les langages

Une machine de Turing avec **semi-oracle** est une machine classique avec en plus un état d'interrogation $q_?$ et un ruban d'oracle. Si \mathcal{L} est un langage, le fonctionnement d'une machine de Turing avec semi-oracle \mathcal{L} est le suivant : sur une entrée m la machine lit m tout en formant un mot sur le ruban d'oracle, jusqu'à entrer dans l'état d'interrogation $q_?$. Si le mot inscrit sur le ruban d'oracle est dans \mathcal{L} alors la machine s'arrête, et continue son calcul sinon.

On définit un ordre sur les langages :

$$\mathcal{L} \preceq \mathcal{L}' \Leftrightarrow \exists \mathcal{M}^{\mathcal{L}'}$$
 une machine de Turing avec semi-oracle \mathcal{L}' telle que $dom(\mathcal{M}^{\mathcal{L}'}) = \mathcal{L}$

où $dom(\mathcal{M})$ est le domaine de la machine \mathcal{M} c'est-à-dire l'ensemble des mots d'entrée sur lesquels le calcul de \mathcal{M} s'arrête.

Propriété 6 La relation \preceq est une relation de pré-ordre.

Démonstration

- \preceq est réflexive : en effet pour tout langage \mathcal{L} on a $\mathcal{L} \preceq \mathcal{L}$. Il suffit de considérer la machine avec oracle \mathcal{L} qui interroge directement l'oracle sur le mot d'entrée.
- \preceq est transitive. Supposons $\mathcal{L}_1 \preceq \mathcal{L}_2$ et $\mathcal{L}_2 \preceq \mathcal{L}_3$ et montrons que $\mathcal{L}_1 \preceq \mathcal{L}_3$. On dispose d'une machine \mathcal{M}_2 avec oracle \mathcal{L}_2 telle que $dom(\mathcal{M}_2) = \mathcal{L}_1$ et d'une machine \mathcal{M}_3 avec oracle \mathcal{L}_3 telle que $dom(\mathcal{M}_3) = \mathcal{L}_2$. On construit une machine \mathcal{M} avec le comportement suivant : sur une entrée m on simule \mathcal{M}_2 sur cette même entrée, et dès que \mathcal{M}_2 veut faire appel à son oracle \mathcal{L}_2 on simule \mathcal{M}_3 . Ainsi \mathcal{M} s'arrête sur m si et seulement si $m \in \mathcal{L}_1$, et c'est bien une machine avec oracle \mathcal{L}_3 . Donc $\mathcal{L}_1 \preceq \mathcal{L}_3$. \square

A partir de ce pré-ordre on définit une relation d'équivalence \approx par :

$$\mathcal{L} \approx \mathcal{L}' \Leftrightarrow \begin{cases} \mathcal{L} \preceq \mathcal{L}' \\ \mathcal{L}' \preceq \mathcal{L} \end{cases}$$

La classe des langages récursivement énumérables est la plus petite pour ce pré-ordre : on a $\emptyset \approx \mathcal{L}$ pour tout langage \mathcal{L} récursivement énumérable.

3.4 Théorème de clôture

L'ordre sur les langages défini par les machines de Turing à semi-oracle correspond à un ordre sur les sous-shifts :

Théorème 3 $Cl_{P,F,SA,SP,TF}(X_{\mathcal{L}}) = \{X_{\mathcal{L}'}/\mathcal{L}' \preceq \mathcal{L}\}$

3.4.1 Inclusion directe

Soit $X_{\mathcal{L}'}$ un sous-shift construit à partir de $X_{\mathcal{L}}$ et d'un nombre fini d'opérations parmi P,F,SA,SP et TF. Pour montrer que $\mathcal{L}' \preceq \mathcal{L}$ on raisonne par induction sur le nombre d'opérations.

Soit $X_{\mathcal{L}''} \subseteq \mathcal{A}^{\mathbb{Z}^d}$ un sous-shift obtenu à partir de $X_{\mathcal{L}}$ par un nombre fini d'opérations parmi P,F,SA,SP et TF. Par hypothèse d'induction, \mathcal{L}'' est tel que $\mathcal{L}'' \preceq \mathcal{L}$. On note \mathcal{M}'' une machine de Turing avec semi-oracle \mathcal{L} telle que $dom(\mathcal{M}'') = \mathcal{L}''$. Si $X_{\mathcal{L}'}$ est obtenu à partir de $X_{\mathcal{L}''}$ en appliquant une opération parmi P,F,SA,SP et TF, montrons que l'on a aussi $\mathcal{L}' \preceq \mathcal{L}$. Pour cela on explique rapidement comment construire une machine \mathcal{M}' avec semi-oracle \mathcal{L} qui s'arrête exactement sur les motifs interdits de \mathcal{L}' .

- **Stabilité par produit**

Il faut un deuxième sous-shift $X_{\mathcal{L}'''}$ sur lequel s'applique aussi l'hypothèse d'induction. Si $m = (m_1, m_2)$ est un motif sur l'alphabet produit, on simule les deux machines \mathcal{M}'' et \mathcal{M}''' (un pas de calcul chacune à tour de rôle) sur les deux projections m_1 et m_2 du motif m .

- **Stabilité par facteur**

Si $\mathcal{L}' = \Pi(\mathcal{L}'')$ où Π est un morphisme : $\mathcal{A}^{\mathbb{Z}^d} \rightarrow \mathcal{B}^{\mathbb{Z}^d}$ et de voisinage \mathbb{V} . Si m est un motif sur l'alphabet \mathcal{B} , on simule la machine \mathcal{M}'' sur chacun des motifs $m' \in \mathcal{A}^{supp(m)+\mathbb{V}}$ tel que $\Pi(m') = m$ en faisant une étape de calcul sur chaque m' à tour de rôle.

- **Stabilité par sous-action**

Si \mathcal{L}' est tel que $X_{\mathcal{L}'} = (X_{\mathcal{L}''})_k^{\mathbb{G}}$ avec $k \in \mathbb{N}$ et $\mathbb{G} < \mathbb{Z}^d$. on simule \mathcal{M}'' sur toutes les façons de compléter le motif à tour de rôle. C'est exactement le même principe que dans la preuve du théorème 2.

- **Stabilité par superposition**

Soit m un motif de $\phi_{SP}(\mathcal{A}^{\mathbb{Z}^d}, \mathbb{Z}^d, \mathbb{Z}^{d'})$, alors m est la superposition de d' motifs de $\mathcal{A}^{\mathbb{Z}^d}$. Il suffit de simuler \mathcal{M}'' sur chacun de ces motifs à tour de rôle.

- **Stabilité par type fini**

On commence par vérifier que le motif ne contient pas de motifs interdits du SFT. Si c'est le cas, on s'arrête sinon on simule \mathcal{M}'' .

3.4.2 Inclusion réciproque

On se donne \mathcal{L} un langage et \mathcal{L}' un autre langage tel que $\mathcal{L}' \preceq \mathcal{L}$. Sans perte de généralité, on peut supposer que \mathcal{L} est de dimension 1. Si ce n'est pas le cas, on s'y ramène de la façon suivante : il existe une bijection récursive de \mathbb{Z}^d dans \mathbb{Z} , donc à tous motif de $\mathcal{E}_{\mathcal{A}}^d$ on peut associer un motif de $\mathcal{E}_{\mathcal{A}}^1$. Donc à un ensemble de motifs interdits dans $\mathcal{E}_{\mathcal{A}}^d$ correspond un ensemble de motifs interdits dans $\mathcal{E}_{\mathcal{A}}^1$ ce qui permet de se ramener à la dimension 1.

On montre ici le résultat dans le cas particulier où \mathcal{L}' est de dimension 1, mais on peut généraliser la preuve à la dimension d . Il s'agit de construire à l'aide des opérations P,F,SA,SP,TF et de $X_{\mathcal{L}}$ le sous-shift $X_{\mathcal{L}'}$.

Comme $\mathcal{L}' \preceq \mathcal{L}$ il existe une machine de Turing \mathcal{M} avec semi-oracle \mathcal{L} telle que $dom(\mathcal{M}^{\mathcal{L}}) = \mathcal{L}'$. On modifie le fonctionnement de cette machine pour que d'une part elle ne prenne en entrée que des motifs de taille 2^n , et que d'autre part au moment où elle entre dans l'état d'interrogation $q_?$ le mot sur le ruban

d'oracle soit recopié dans l'alphabet $\tilde{\mathcal{A}}$, une copie de \mathcal{A} , puis à nouveau dans l'alphabet \mathcal{A} une fois que l'oracle a été appelé.

Tout d'abord la liste de tous les sous-shifts qu'on utilisera pour construire $X_{\mathcal{L}'}$:

- Le sous-shift de départ $X_{\mathcal{L}}$ que l'on écrit dans une copie de \mathcal{A} : $\tilde{X}_{\mathcal{L}} \subseteq \tilde{\mathcal{A}}^{\mathbb{Z}}$ servira à simuler l'oracle.
- On peut coder le fonctionnement de \mathcal{M} par un SFT $\Sigma_{\mathcal{M}} \subseteq \mathcal{O}^{\mathbb{Z}^2}$, où \mathcal{O} est un alphabet contenant \mathcal{A} , $\tilde{\mathcal{A}}$ et \mathcal{B} .
- Pour construire les espaces de calcul on utilisera Σ_2 , Σ_3 et Σ_5 défini dans partie 1.4.1 ; ce sont tous trois des SFT définis sur l'alphabet $\{\bullet, \circ\}$.

Construction du sous-shift $X_{\mathcal{L}'}$

Pour construire $X_{\mathcal{L}'}$ on va construire un sous-shift $\Sigma' \in \mathcal{Cl}_{P,F,SA,SP,TF}(X_{\mathcal{L}})$ de dimension 4 sur un alphabet \mathcal{C} qui sera précisé dans la suite. On appelle (e_1, e_2, e_3, e_4) la base canonique de \mathbb{Z}^4 . Les quatre dimensions de Σ' seront utilisées de la façon suivante :

- on construit le sous-shift $X_{\mathcal{L}'}$ sur $\mathbb{Z}e_1$
- grâce à $\mathbb{Z}e_1 + \mathbb{Z}e_2 + \mathbb{Z}e_3$ on construit les rectangles pour simuler \mathcal{M} , chaque rectangle étant dans un plan $\{i\} \times \mathbb{Z} \times \mathbb{Z} \times \{k\}$
- sur $\mathbb{Z}e_4$ on écrit le sous-shift $X_{\mathcal{L}}$ qui servira à simuler le langage oracle \mathcal{L}

Etape 1 : On remarque tout d'abord qu'on peut transformer $X_{\mathcal{L}}$ en $\tilde{X}_{\mathcal{L}}$ par un simple morphisme lettre à lettre. On construit ensuite $\tilde{W} = \phi_{SP}(\tilde{X}_{\mathcal{L}}, \mathbb{Z}e_4, \mathbb{Z}e_1 + \mathbb{Z}e_2 + \mathbb{Z}e_3)$ pour placer $\tilde{X}_{\mathcal{L}}$ dans un sous-shift de dimension 4. On ajoute ensuite par produit les lettres de l'alphabet \mathcal{C} : $W = \tilde{W} \times \mathcal{C}'^{\mathbb{Z}^4}$ de sorte que $W \subseteq \mathcal{C}^{\mathbb{Z}^4}$ et $W \in \mathcal{Cl}_{P,F,SP}(X_{\mathcal{L}})$.

Etape 1bis : On veut que $X_{\mathcal{L}'}$ apparaisse sur $\mathbb{Z}e_1$. Les calculs pour simuler la machine \mathcal{M} prendront comme mot d'entrée un motif situé sur $\mathbb{Z}e_2$. Il faut donc recopier ce qui est sur $\mathbb{Z}e_1$ sur $\mathbb{Z}e_2$ pour que les calculs de la machine portent sur ce qui sera le sous-shift $X_{\mathcal{L}'}$. On le fait par la condition de type fini :

$$\forall x \in \mathcal{C}^{\mathbb{Z}^4}, \forall u \in \mathbb{Z}^4, x_u = x_{u+e_1-e_2}$$

On veut aussi que sur chaque espace de calcul, le mot d'entrée reste accessible tout au long du calcul de la machine. On ajoute pour cela la condition :

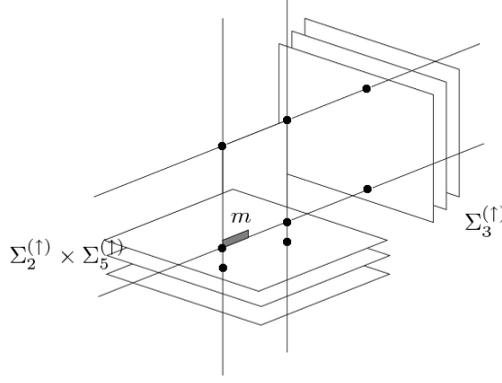
$$\forall x \in \mathcal{C}^{\mathbb{Z}^4}, \forall u \in \mathbb{Z}^4, x_u = x_{u+e_3}$$

On obtient un sous-shift $W' \in \mathcal{Cl}_{P,F,SP,TF}(X_{\mathcal{L}})$.

Etape 2 : On ajoute à W' des espaces de calcul. Pour cela, on construit séparément un SFT \tilde{W}_{rect} qui contiendra des rectangles bien choisis :

- on note F_i l'ensemble fini de motifs qui assure que $\forall x \in \{\bullet, \circ\}^{\mathbb{Z}^3}, \forall u \in \mathbb{Z}^3, x_u = x_{u+e_i}$
- $W_{2,5} = \phi_{TF}(\phi_{SP}(\Sigma_2^{(\dagger)} \times \Sigma_5^{(\dagger)}, \mathbb{Z}e_1 + \mathbb{Z}e_2, \mathbb{Z}e_3), F_3)$
- $W_3 = \phi_{TF}(\phi_{SP}(\Sigma_3^{(\dagger)}, \mathbb{Z}e_1 + \mathbb{Z}e_3, \mathbb{Z}e_2), F_2)$
- $\tilde{W}_{rect} = W_{2,5} \times W_3$

On obtient une structure qui ressemble à celle décrite en 1.4.3, sauf que les bases des rectangles contiennent deux informations : la taille effective de la base, donnée par Σ_5 , et la taille de l'entrée donnée par Σ_2 . De cette manière, on pourra simuler des calculs de machine de Turing avec une taille de l'entrée de 2^n , une taille du ruban limitée à 5^m et 3^p étapes de calculs.



Le sous-shift \tilde{W}_{rect} ainsi obtenu est un SFT donc il existe un ensemble fini de motifs F_{rect} tel que $\tilde{W}_{rect} = X_{F_{rect}}$. On ajoute cette structure à W' via l'opération $W_{rect} = \phi_{TF}(W', F_{rect})$. On a bien $W_{rect} \in \mathcal{Cl}_{P,F,SP,TF}(X_{\mathcal{L}})$.

Etape 3 : On fait fonctionner la machine \mathcal{M} à l'intérieur des rectangles, pour le moment sans tenir compte des appels à l'oracle. Pour cela on construit $\tilde{W}_{\mathcal{M}} = \Sigma_{\mathcal{M}} \times W_{rect}$ auquel on ajoute les conditions de type fini F_{calc} qui permettent de placer une seule tête de lecture au coin inférieur gauche de chaque rectangle et de stopper un calcul lorsqu'on atteint les bords du rectangle. Pour l'instant à chaque appel à l'oracle la machine continue son calcul. Ainsi $W_{\mathcal{M}} = \phi_{TF}(\tilde{W}_{\mathcal{M}}, F_{calc}) \in \mathcal{Cl}_{P,F,SP,TF}(X_{\mathcal{L}})$

Etape 4 : On simule l'oracle. Pour cela on ajoute des conditions de type fini pour s'assurer que lorsqu'au cours d'un calcul, la machine fait appel à l'oracle en $(i, j, k, l) \in \mathbb{Z}^4$, le motif $m \in \tilde{\mathcal{A}}^n$ sur lequel est interrogé l'oracle coïncide avec le motif situé sur $\mathbb{Z}e_4$ entre (i, j, k, l) et $(i, j, k, l+n)$. Les motifs autorisés sont de la forme :

$$\begin{array}{c} \uparrow e_4 \\ \begin{array}{|c|c|} \hline \tilde{a} & \cdot \\ \hline \tilde{b} & \tilde{a} \\ \hline \end{array}, \begin{array}{|c|c|} \hline \tilde{a} & \cdot \\ \hline (q?, \tilde{b}) & \tilde{a} \\ \hline \end{array} \\ \rightarrow e_2 \end{array}$$

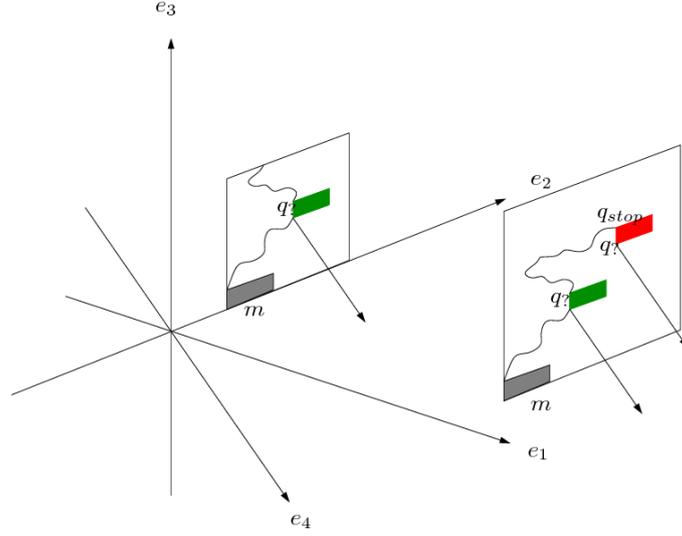
On note ces conditions de type fini F_{oracle} . On a donc $W_{\mathcal{M}_{oracle}} = \phi_{TF}(W_{\mathcal{M}}, F_{oracle}) \in \mathcal{Cl}_{P,F,SP,TF}(X_{\mathcal{L}})$.

Etape 5 : On fait en sorte que chaque configuration de $X_{\mathcal{L}}$ apparaissant sur \mathbb{Z}^4 soit utilisée pour les mêmes calculs, afin d'éviter les problèmes de dépendance entre les calculs. Ceci est fait grâce à la condition de type fini :

$$\forall x \in \mathcal{C}^{\mathbb{Z}^4}, \forall u \in \mathbb{Z}^4, x_u = x_{u+e_1+e_4}$$

Enfin on interdit l'état d'arrêt q_{stop} de la machine \mathcal{M} . On note Σ le sous-shift ainsi obtenu. On a bien $\Sigma \in \mathcal{Cl}_{P,F,SP,TF}(X_{\mathcal{L}})$.

Le dessin ci-dessous résume le fonctionnement de \mathcal{M} sur les rectangles ainsi construits :



Sur un motif m de taille 2^n on simule le fonctionnement de la machine \mathcal{M} . Dès que celle-ci fait appel à l'oracle on compare le mot sur lequel l'oracle est appelé avec celui sur $\mathbb{Z}e_4$. Si les deux mots coïncident on continue le calcul, sinon on entre dans l'état d'arrêt q_{stop} . Si la machine ne peut pas finir le calcul dans le temps donné par la hauteur du rectangle, on peut toujours trouver un rectangle plus haut qui calculera sur la même entrée.

La construction est correcte Montrons que le sous-shift $\Sigma_0^{\mathbb{Z}e_1}$ qu'on obtient en prenant la projection du sous-shift précédent sur $\mathbb{Z}e_1$ est bien $X_{\mathcal{L}'}$, à un facteur près qui consiste à ne garder que l'information correspondant à l'alphabet \mathcal{B} .

- $\Sigma_1^{\mathbb{Z}e_1} \subseteq X_{\mathcal{L}'}$

Soit $y \in \Sigma$, montrons que $x = y|_{\mathbb{Z}e_1} \in X_{\mathcal{L}'}$. Il suffit pour cela de montrer que tout motif qui apparaît dans x n'est pas dans \mathcal{L}' . Soit m un motif qui apparaît dans x , alors c'est un sous-motif d'un motif \tilde{m} de x de taille 2^n . Alors pour tout $t \in \mathbb{N}$ et $s \in \mathbb{N}$, il existe un rectangle de hauteur au moins t et de largeur au moins s avec \tilde{m} comme mot d'entrée. Comme $y \in \Sigma$, les calculs de la machine \mathcal{M} sur chacun de ces rectangles de taille arbitrairement grande n'aboutissent jamais à l'état d'arrêt q_{stop} , ce qui signifie que la machine \mathcal{M} boucle sur l'entrée \tilde{m} : on a bien $m \notin \mathcal{L}'$.

- $X_{\mathcal{L}'} \subseteq \Sigma_1^{\mathbb{Z}e_1}$

Soit $x \in X_{\mathcal{L}'}$, montrons que l'on peut construire une configuration $y \in \mathcal{C}^{\mathbb{Z}^4}$ avec les contraintes :

- $y \in \Sigma$
- $y|_{\mathbb{Z}e_1} = x$

Pour avoir $y \in \Sigma$ il suffit de vérifier que pour tout $(i, j, k) \in \mathbb{Z}^3$, on peut imposer $y_{\{i\} \times \{j\} \times \{k\} \times \mathbb{Z}} \in X_{\mathcal{L}}$ tout en s'assurant que les calculs de la machine \mathcal{M} sur les rectangles ne conduisent pas à l'état q_{stop} .

Regardons un espace de calcul (un rectangle). Alors dans cet espace on calcule sur l'entrée m , un motif de taille 2^n qui apparaît dans x . Comme m apparaît dans x , $m \notin \mathcal{L}'$ donc la machine \mathcal{M} boucle sur l'entrée m . Cela signifie qu'à chaque fois que le calcul de \mathcal{M} sur m fait appel à l'oracle sur un motif m' , celui-ci n'est pas dans \mathcal{L} . Il existe donc pour tout motif m' sur lequel on interroge l'oracle une configuration $z \in X_{\mathcal{L}}$ telle que $z_{[0;|m'|-1]} = m'$. Il suffit de compléter y de la manière suivante :

- si en $(i, j, k) \in \mathbb{Z}^3$ le calcul de \mathcal{M} fait appel à l'oracle sur un motif m' , alors $y_{\{i\} \times \{j\} \times \{k\} \times \mathbb{Z}} = z$ construit précédemment ;
- sinon si le calcul ne fait pas appel à l'oracle, on complète avec $y_{\{i\} \times \{j\} \times \{k\} \times \mathbb{Z}} \in X_{\mathcal{L}}$ quelconque.

Ceci assure que y est bien dans le sous-shift Σ , donc $x \in \Sigma_1^{\mathbb{Z}e_1}$.

Ceci achève la démonstration. \square

Question : Est-il possible de construire un sous-shift de dimension 3 seulement ? On ne peut pas se contenter faire le produit avec le sous-shift $X_{\mathcal{L}}$ selon la direction $\mathbb{Z}e_2$, car cela entraînerait des problèmes

de synchronisation, comme le montre l'exemple suivant.

Si $X_{\mathcal{L}}$ est $\{a^\infty ba^\infty, a^\infty\} \subseteq \{a, b\}^{\mathbb{Z}}$ et que dans un même plan $\{i\} \times \mathbb{Z}^2 \times \{l\}$ on fait appel à l'oracle avec pour mot d'entrée aba en (i, j, k, l) sur un premier rectangle et aba en $(i, j + j', k, l)$ sur un second, on ne peut pas trouver d'élément du sous-shift $X_{\mathcal{L}}$ qui fasse en sorte que la machine continue ses calculs sur les deux rectangles.

Conclusion et perspectives

L'approche dynamique développée ici à l'aide d'opérations sur les sous-shifts donne de nouveaux outils pour étudier le cas multi-dimensionnel. L'objectif est de comprendre la structure des pré-ordres ainsi définis. Pour cela plusieurs pistes sont envisageables.

Une amélioration possible du théorème 3 serait de diminuer la dimension (ici $d+3$ pour construire un sous-shift de dimension d) dont on a besoin. Pour affiner le résultat on peut aussi chercher à caractériser des ordres obtenus en interdisant les opérations qui semblent essentielles au théorème 3. Qu'obtient-on à partir d'un sous-shift donné sans l'opération de type fini TF ? Enfin une question difficile encore non résolue est celle de l'existence de sous-shifts universels, c'est-à-dire des sous-shifts particuliers qui sont plus grands, pour un ordre défini les opérations, que toute une classe de sous-shifts. Dans [Hoc07a] il est montré que de tels sous-shifts existent en dimension 1 pour la classe des sous-shifts récursivement énumérables, mais qu'il n'en existe pas pour cette même classe en dimension 2. Cependant ses résultats concernent des sous-shifts beaucoup plus généraux ayant pour alphabet l'ensemble de Cantor $\{0, 1\}^{\mathbb{N}}$.

Références

- [Bea93] M.P. Beal. Codage Symbolique, 1993.
- [Ber66] R. Berger. *The Undecidability of the Domino Problem*. American Mathematical Society, 1966.
- [BL97] M. Boyle and D. Lind. Expansive Subdynamics. *Transactions of the American Mathematical Society*, 349(1) :55–102, 1997.
- [Hed69] GA Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Theory of Computing Systems*, 3(4) :320–375, 1969.
- [HM07] M. Hochman and T. Meyerovitch. A Characterization of the Entropies of Multidimensional Shifts of Finite Type. *Arxiv preprint math.DS/0703206*, 2007.
- [Hoc07a] M. Hochman. A Note on Universality in Multidimensional Symbolic Dynamics. 2007.
- [Hoc07b] M. Hochman. On the Dynamics and Recursive Properties of Multidimensional Symbolic Systems. 2007.
- [Kit98] B. Kitchens. *Symbolic dynamics*. Springer New York, 1998.
- [Kur03] P. Kurka. Topological and Symbolic Dynamics. *Societe Mathematique de France, Paris*, 2003.
- [LM95] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1995.
- [Moz89] S. Mozes. Tilings, substitution systems and dynamical systems generated by them. *Journal d'analyse mathématique(Jerusalem)*, 53 :139–186, 1989.
- [MR99] J. Mazoyer and I. Rapaport. Inducing an order on cellular automata by a grouping operation. *Discrete Applied Mathematics*, 91(1-3) :177–196, 1999.
- [Oll02] N. Ollinger. *Automates Cellulaires : structures*. PhD thesis, École Normale Supérieure de Lyon, 2002.
- [RJ87] H. Rogers Jr. *Theory of recursive functions and effective computability*. MIT Press Cambridge, MA, USA, 1987.
- [Rob71] R.M. Robinson. Undecidability and nonperiodicity for tilings of the plane. *Inventiones Mathematicae*, 12(3) :177–209, 1971.
- [The05] G. Theyssier. *Automates cellulaires : un modèle de complexités*. PhD thesis, École Normale Supérieure de Lyon, 2005.