

AIX-MARSEILLE UNIVERSITÉ
ÉCOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE E.D. 184

THÈSE

présentée pour obtenir le grade de
DOCTEUR D'AIX-MARSEILLE UNIVERSITÉ
Spécialité : Informatique

par

Pascal Vanier

sous la direction de Emmanuel JEANDEL

Titre :

Pavages : périodicité et complexité calculatoire

soutenue publiquement le 22 novembre 2012

JURY

Marie-Pierre BÉAL	Université de Marne-la-Vallée	<i>Rapporteur</i>
Julien CASSAIGNE	CNRS & Aix-Marseille Université	<i>Membre</i>
Bruno DURAND	Université Montpellier II	<i>Membre</i>
Enrico FORMENTI	Université de Nice Sophia-Antipolis	<i>Membre</i>
Emmanuel JEANDEL	Université de Lorraine	<i>Directeur</i>
Anne SIEGEL	CNRS & Université Rennes I	<i>Membre</i>
Nikolay VERESHCHAGIN	Université d'État de Moscou	<i>Rapporteur</i>

Table des matières

Remerciements	v
Notations	vii
Introduction	ix
1 Définitions préliminaires	1
1.1 Sous-shifts	3
1.1.1 Définitions de base	3
1.1.2 SFTs et tuiles de Wang	6
1.1.3 Quelques opérations sur les sous-shifts	8
1.1.4 Aspects topologiques des sous-shifts	9
1.1.5 Autres types de sous-shifts	11
1.1.6 Sous-shifts de dimension 1	15
1.1.7 Périodicité, quasipériodicité et minimalité	17
1.1.8 Conjugaison et invariants de conjugaison	19
1.2 Calculabilité	22
1.2.1 Réductions et degrés Turing	23
1.2.2 Hiérarchie arithmétique	24
1.2.3 Indécidabilité de la pavabilité du plan	25
2 Classes Π_1^0 et sous-shifts	33
2.1 Classes Π_1^0	34
2.1.1 Degrés Medvedev et Muchnik	35
2.1.2 Des classes Π_1^0 aux pavages	37
2.1.3 Degrés Medvedev des SFTs	37
2.2 Degrés Turing	38
2.2.1 Une grille irrégulière	38
2.2.2 Degrés Turing des classes Π_1^0 et des SFTs	44
3 Caractérisation des ensembles de périodes des SFTs	53
3.1 Diverses notions de périodicité	55
3.2 Périodes horizontales	57
3.3 Fortes périodes	62
3.4 Nombre de points fortement périodiques	69
3.5 1-périodes	72
3.6 Ensembles de Périodes des sous-shifts sofiques et effectifs	77
3.7 Remarques sur les périodes en tant que langages	80

4	Difficulté de la factorisation et de la conjugaison	83
4.1	Conjugaison et égalité	84
4.1.1	Sous-shifts de type fini	84
4.1.2	Sous-shifts effectifs et sofiques	87
4.2	Factorisation	89
4.2.1	La factorisation est Σ_3^0	90
4.2.2	La factorisation est Σ_3^0 -dure	90
5	Bibliographie	97
5.1	Bibliographie personnelle	97
5.2	Bibliographie	97

Remerciements

Sans cette page de remerciements cette thèse serait incomplète, c'est pourquoi il était indispensable que je l'ajoute, surtout maintenant le fameux diplôme en poche. De plus, je dois avouer que c'est la partie la plus agréable de la rédaction et qu'il n'est pas besoin de se forcer pour remercier tous les acteurs qui ont contribué à cet ouvrage, d'une manière ou d'une autre.

Pour commencer par une touche classique, je souhaiterais remercier mon directeur de thèse, Emmanuel, qui a toujours été là pour répondre à mes questions (souvent stupides) et qui a su me guider tout au long de ces trois ans.

Je tiens également à remercier Jarkko, qui m'a accepté en stage il y a maintenant 5 ans et par qui tout a commencé.

Je remercie Marie-Pierre Béal et Nikolay Vereshchagin, qui ont accepté d'être les rapporteurs de ma thèse, ainsi qu'Anne-Siegel, qui a accepté de présider le jury, ainsi que Julien Cassaigne, Bruno Durand et Enrico Formenti qui ont accepté d'en être membres.

Il y a tant d'autres personnes qui a un moment ou à un autre ont contribué au parcours qui m'a mené à la recherche, mais il y a aussi une entité : l'équipe ESCAPE(D), l'équipe dans laquelle j'ai commencé ma thèse, et dans laquelle je l'ai finie. Ainsi, merci à Bruno, pour avoir porté sur ses épaules le lourd poids des charges administratives, à Victor, pour le billard et les histoires, à Nicolas pour les conversations plus ou moins scientifiques (plutôt plus que moins), à Greg, pour son étonnant sens de l'humour et à Sasha, pour son insistance à utiliser le tableau noir. Ceux de l'équipe que j'ai le plus côtoyés sont bien entendu les autres thésards : Alexis, pour les longues conversations pleines de sens sur la guarrigue, Martin, parce qu'il ne rigole pas, Tarik, pour le beat et la mosell'attitude, Fabien, parce que non, on est pas toujours le soir, et Alex, qui était là aussi. Je vais aussi remercier Jérémie dans ce paragraphe, après tout, il faisait presque partie de l'équipe.

Merci aussi aux personnes de la communauté avec qui j'ai discuté (parfois autour d'une bière) et qui m'ont permis d'accroître ma connaissance du domaine : Pierre G., Nathalie, Guillaume, Gaëtan, Laurent, Thomas, Antoine, Timo, Jacques, et bien d'autres !

Mais les personnes qui ont contribué à cette thèse ne s'arrêtent pas là, ainsi, je souhaite également remercier ces camarades avec qui j'ai passé des moments mémorables, que ce soit avant, pendant ou après la thèse. Je pense à Florent, BoolS, Bruno, Rimka, Frood, Yo, Alvaro, Dams, Schag, Pierre B., et les potes d'escalade.

Merci aussi à mes parents, à la fois pour leur rôle de mécènes de la science et pour leur présence, si rassurante. Je vois remercier Jean-Michel et Benoit pour des raisons fallacieuses : Jean-Michel, pour avoir fait de la vie quotidienne

un combat et Benoit pour l'inverse.

Et surtout, surtout, je remercie Alexandra, qui a été là pendant ces trois ans, qui m'a soutenu dans les moments de doute, dont la confiance me galvanise chaque jour et sans qui la préparation de cette thèse n'aurait pas pu être aussi agréable.

Notations

$\mathcal{X}_{\mathcal{F}}$	Le sous-shift défini par la collection de motifs interdits \mathcal{F} (4)
Σ_X	alphabet du sous-shift X . (4)
$\mathcal{A}_n(\mathcal{F}), \mathcal{E}_n(\mathcal{F})$	Ensemble des motifs admissibles et extensibles de $\mathcal{X}_{\mathcal{F}}$ de support \mathcal{B}_n (4)
\mathcal{B}_n	La boule $\llbracket -n, n \rrbracket^d$ (4)
$\sigma_{\mathbf{z}}$	Opérateur de translation par $\mathbf{z} \in \mathbb{Z}^d$ (4)
$\ \mathbf{z}\ _{\infty}$	Norme infinie, si $\mathbf{z} = (z_1, \dots, z_d)$, alors $\ \mathbf{z}\ _{\infty} = \max_i z_i $. (9)
\mathbf{e}_i	Base de \mathbb{Z}^d : $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$ (10)
$\psi(x)\downarrow, \psi(x)\downarrow^t, \psi(x)\uparrow$	ψ s'arrête (s'arrête en t étapes, ne s'arrête pas) sur l'entrée x (22)
\leq_T, \equiv_T, \deg_T	Réduction, équivalence et degré Turing. (23)
\leq_m	Réduction many-one. (23)
Γ_x	$\{\mathbf{v} \in \mathbb{Z}^d \mid \forall \mathbf{z} \in \mathbb{Z}^d, x_{\mathbf{z}+\mathbf{v}} = x_{\mathbf{z}}\}$, le treillis des vecteurs de périodicité de x
$\mathfrak{P}_X, \mathfrak{P}_X^1, \mathfrak{P}_X^h$	Ensembles des : périodes fortes, 1-périodes, périodes horizontales (55)

Introduction

Historiquement, les pavages ont été introduits par deux communautés distinctes, ayant deux points de vue et donc deux formalismes différents.

Le premier de ces formalismes, les tuiles de Wang [Wan61], a été introduit afin de déterminer l'indécidabilité de certains problèmes de logique, voir Kahr, Moore et Wang [KMW62]. Ce sont des tuiles carrées dont les bords sont colorés et un jeu de tuiles est un ensemble fini de tuiles de Wang. Un pavage par un jeu de tuiles est une décoration du plan \mathbb{Z}^2 , par les tuiles du jeu, telle que deux tuiles voisines ont leur bord en commun de la même couleur. Ce formalisme a également permis de montrer de nombreux résultats d'indécidabilité dans le domaine des automates cellulaires, voir Kari [Kar92, Kar94, Kar90b, Kar90a].

Le second de ces formalismes, les sous-shifts de type fini (SFTs), a été introduit par des dynamiciens, Morse [Mor21], Morse et Hedlund [MH38], afin d'étudier le comportement de certains systèmes dynamiques. Ils ont également été utilisés par Shannon [Sha48] comme modèle pour les canaux de communication discrets ou encore par IBM pour le codage d'informations sur certains supports magnétiques. On peut définir de manière générale les sous-shifts de dimension d , qui sont les coloriage de \mathbb{Z}^d par un nombre fini de couleurs ne contenant aucun motif interdit par une famille spécifiée. Ils sont de type fini lorsque cette famille de motifs interdits est finie. Ils peuvent également être définis de manière purement topologique : étant donné un ensemble de couleurs Σ , ce sont les sous-ensembles fermés et invariants par translation de $\Sigma^{\mathbb{Z}^d}$.

Le formalisme que nous adopterons ici est celui des sous-shifts, qui est légèrement plus général et qui permet de définir des sous-classes intéressantes : les sous-shifts effectifs, qui sont des sous-shifts que l'on peut définir avec un ensemble de motifs interdits récursivement énumérable, et les sous-shifts sofiques, qui sont les projections lettre à lettre des SFTs.

En dimension 1, les SFTs forment un domaine de recherche très riche. La pléthore de résultats, voir Lind et Marcus [LM95], est en partie due au fait qu'à chaque SFT correspond un graphe, les points du SFT étant les marches bi-infinies sur ce graphe. De ce fait, l'étude de ceux-ci entretient des liens très forts avec la théorie des automates, mais aussi avec l'algèbre linéaire. La plupart des problèmes y sont décidables, à commencer par savoir si un SFT est non-vide : le problème revient à déterminer si le graphe qui le représente contient un cycle, donc de manière équivalente à savoir si le SFT contient un point périodique.

À partir de la dimension 2, il n'y a plus de liens simples avec les graphes, Nasu [Nas95], et les SFTs deviennent plus complexes : par exemple, le problème de savoir si un SFT est non-vide devient indécidable, comme cela a été prouvé par Berger [Ber64, Ber66]. Ce résultat est basé sur la construction d'un jeu de

tuiles apériodique : un jeu de tuiles qui ne produit que des pavages apériodiques. Des SFTs ne contenant que des points apériodiques ne pouvaient pas exister en dimension 1. Ceci laisse présager que la plupart des propriétés sur les SFTs en dimensions supérieures à 2 seront indécidables, et peut donc rendre pessimiste. Lind [Lin04] en dit d’ailleurs :

The fact that none of these three basic questions, (1) the existence of points, (2) the extension of finite configurations, (3) the existence of periodic points, can be decided by a finite procedure is what I call the “Swamp of Undecidability.” It’s a place you don’t want to go.

Dans cette thèse, nous sautons avec joie et allégresse dans ce “bourbier” : nous considérons la théorie de la calculabilité et de la complexité comme un outil servant à l’étude des SFTs de dimension supérieure à 2. Cette approche est motivée par le fait que les pavages sous leurs différentes formes ont permis d’exhiber des problèmes complets simples pour de nombreuses classes de calculabilité et de complexité grâce à un codage très simple des diagrammes espace-temps de machines de Turing. Ainsi, les pavages ont permis à Levin [Lev73] de montrer l’existence d’un premier problème **NP**-complet en même temps que Cook et d’un problème **PSPACE**-complet. Ils ont aussi servi à trouver des problèmes **NP**-complets en moyenne, Levin [Lev84, Lev86], Π_1^0 et Σ_1^0 -complets, Berger [Ber64], Π_1^1 -complets et Σ_1^1 -complets, Harel [Har85]. Emde Boas [Emd97] a même montré que de nombreuses réductions en complexité étaient plus simples lorsque l’on parlait d’un problème de pavage. Nous allons donc voir la calculabilité et la complexité comme un outil servant à étudier les sous-shifts plutôt que comme un obstacle insurmontable, dans la lignée de certains résultats liant la calculabilité et l’étude des sous-shifts [AS09, Hoc09a, HM10, Mey10, Sim11b].

Le problème qui constitue le fil directeur de cette thèse et que nous attaquons sous cet angle est la conjugaison, un problème lié à l’aspect dynamique des SFTs. Deux sous-shifts sont conjugués quand il existe un codage local réversible qui transforme toute configuration de l’un en une configuration de l’autre : chacun des deux sous-shifts code l’autre. La conjugaison est la bonne notion d’isomorphisme pour les sous-shifts et peut être vue comme une relation d’équivalence qui permet de séparer les sous-shifts en classes d’équivalence. Les sous-shifts d’une même classe sont du même type (fini, effectif, sofique...) et ont exactement le même comportement, les mêmes propriétés dynamiques. Un sous-shift X peut également coder un sous-shift Y sans que cela soit réciproque, on dit alors que X se factorise sur Y . La dynamique de Y est alors une “sous-dynamique” de X .

Le problème de la conjugaison pour les SFTs et sous-shifts soifiques est décidable sur \mathbb{N} , voir Williams [Wil73], indécidable sur \mathbb{N}^d et \mathbb{Z}^d pour $d \geq 2$ et ouvert sur \mathbb{Z} . Pour les sous-shifts effectifs il est indécidable sur \mathbb{N}^d et \mathbb{Z}^d pour toute dimension $d \geq 1$. Le problème de la factorisation est indécidable en dimensions $d \geq 2$ et ouvert sur \mathbb{Z} , bien qu’une réponse partielle ait été donnée par Boyle [Boy83]. Même si l’on sait que ces deux problèmes sont indécidables en dimensions $d \geq 2$, connaître leur degré de difficulté exact donnerait peut-être une intuition sur leur difficulté en dimension 1.

Comme on ne peut pas répondre en général au problème de la conjugaison, des stratégies ont été développées pour pouvoir répondre dans certains cas. Les invariants de conjugaisons font partie de ces stratégies et permettent parfois de montrer que deux sous-shifts ne sont pas conjugués. Un invariant de conjugai-

son est un objet associé à un sous-shift qui sera le même pour deux sous-shifts conjugués : cela peut être par exemple un nombre réel, un groupe ou encore une suite. Les deux invariants les plus étudiés sont l'entropie, qui est un nombre réel mesurant la croissance du nombre de motifs en fonction de leur taille, et le nombre de points périodiques, qui est la suite $(p_n)_{n \in \mathbb{N}^*}$ où p_n est le nombre de points de période n du sous-shift. Ces deux invariants n'ont pas qu'un intérêt pour la séparation des classes de sous-shifts, ils permettent également de mieux comprendre la dynamique de ceux-ci. C'est pourquoi il est intéressant de savoir quelles valeurs ils peuvent prendre. En dimension 1 ces deux invariants ont été caractérisés entièrement à l'aide de quantités liées à la matrice les représentant. En dimensions supérieures, seule l'entropie a fait l'objet d'une caractérisation par Hochman et Meyerovitch [HM10] : les entropies de SFTs sont exactement les réels récursivement énumérable par la droite. Cette caractérisation a ensuite été généralisée à d'autres invariants de croissance du nombre de motifs par Meyerovitch [Mey10], dont les caractérisations étaient encore de nature calculatoire. Dans cet article, Meyerovitch a posé la question de la caractérisation du nombre de points périodiques en dimensions $d \geq 2$, suggérant que le calcul pourrait, une fois de plus, jouer un rôle important.

Ce rôle du calcul dans l'étude des SFTs est finalement assez naturel étant donné le codage très simple des diagrammes espace-temps de machines de Turing. Toutefois, une étude plus approfondie du lien entre les puissances de calcul des SFTs et des machines de Turing permet de mieux comprendre les SFTs en tant que modèle de calcul. Les SFTs sont des classes Π_1^0 de $\{0, 1\}^{\mathbb{N}}$, un objet classique et bien compris de la théorie de la calculabilité, voir Cenzer et Remmel [CR98] par exemple. La compréhension de la structure calculatoire des SFTs passe donc par la comparaison de ceux-ci aux classes Π_1^0 . La question la plus naturelle est de savoir si toute classe Π_1^0 est récursivement isomorphe à un SFT. En effet, les résultats prouvés jusqu'à maintenant suggèrent un lien très fort entre les deux objets, Simpson [Sim11b] a par exemple démontré que les SFTs avaient exactement les mêmes degrés Muchnik et Medvedev que les classes Π_1^0 . Miller [Mil12] a quant à lui prouvé le même résultat pour les sous-shifts effectifs en dimension 1. Les degrés Medvedev et Muchnik, sont des mesures de la complexité des points les plus simples d'une classe Π_1^0 et à ce titre ne donnent qu'une information partielle sur la similitude entre les sous-shifts de type fini ou effectifs et les classes Π_1^0 . En particulier, ils ne donnent pas toute la structure calculatoire de l'objet. Pour connaître celle-ci plus en détail, il faut s'intéresser, étant donné un SFT, à l'ensemble des degrés Turing de ses points. Cet ensemble permet une meilleure compréhension de la structure calculatoire d'une classe Π_1^0 , mais n'est pas aisément manipulable. Cenzer, Dashti et King [CDK08], Cenzer, Dashti, Toska et Wyman [Cen+10, Cen+12] ont montré que les ensembles de degrés Turing des classes Π_1^0 dénombrables et des sous-shifts effectifs dénombrables étaient les mêmes. La comparaison entre les degrés Turing des SFTs et des classes Π_1^0 pourrait également permettre de prouver qu'il n'existe pas d'isomorphisme récursif entre les classes Π_1^0 et les SFTs en général. Par ailleurs, comme la conjugaison est une forme particulière d'isomorphisme récursif, les degrés Medvedev, les degrés Muchnik et l'ensemble des degrés Turing associés à un sous-shift sont invariants par conjugaison.

Contenu de la thèse

Dans cette thèse, nous tentons d'apporter une réponse à toutes ces questions. Ainsi nous commencerons au chapitre 1 par définir les objets que nous étudions. Nous présentons dans celui-ci certaines preuves illustrant des méthodes utilisées dans l'étude des sous-shifts ou en calculabilité, ainsi que la construction de Robinson [Rob71] dont nous ferons usage à plusieurs reprises par la suite.

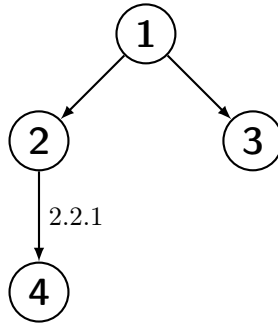
Le chapitre 2 sera consacré à l'étude de la structure calculatoire des sous-shifts et à sa comparaison avec celle des classes Π_1^0 . Nous montrons qu'il existe un quasi-isomorphisme récursif entre les classes Π_1^0 et les SFTs : pour toute classe Π_1^0 S , il existe un sous-shift et un ensemble récursif C de points récursifs de X tels que S et $X \setminus C$ soient isomorphes. Ceci permet de prouver que les classes Π_1^0 contenant un point récursif et les SFTs contenant un point récursif ont exactement les mêmes ensembles de degrés Turing. Étant donné que les classes Π_1^0 dénombrables contiennent toujours un point récursif, ceci permet de compléter les résultats des articles [CDK08, Cen+10, Cen+12] sur le cas dénombrable. Nous prouvons également dans ce chapitre un résultat sur la structure de l'ensemble de degrés Turing des SFTs ne contenant aucun point récursif : ceux-ci sont de mesure de Martin 1, voir Kechris [Kec95]. C'est-à-dire qu'ils admettent toujours un cône de degrés Turing : un degré Turing et tous les degrés supérieurs à celui-ci. Ce résultat structurel permet de montrer qu'il n'existe pas en général d'isomorphisme récursif entre les classes Π_1^0 et les SFTs, et donc en particulier que l'on ne peut pas obtenir mieux qu'un quasi-isomorphisme.

Dans chapitre 3 nous nous intéressons aux ensembles de périodes des sous-shifts de type fini, effectifs et sofiqes. Contrairement à la dimension 1, en dimensions supérieures, il peut y avoir de nombreuses notions de périodicité, que nous définirons dans un premier temps. La définition de périodicité la plus intéressante et proche de celle de la dimension 1 que nous étudions est la périodicité forte, c'est à dire quand le treillis des vecteurs de périodicité est de la forme $n\mathbb{Z}^d$. Pour les sous-shifts de type fini de toutes dimensions $d \geq 2$, nous caractérisons les treillis de cette forme pouvant apparaître dans les points à l'aide de la classe de complexité **NE**. Nous parvenons également à caractériser le nombre de points fortement périodiques à l'aide de la classe de complexité de comptage **#E**. Cependant, ces deux caractérisations nécessitent de considérer les SFTs en *toutes* dimensions. Nous caractérisons également une autre notion de périodicité, cette fois-ci uniquement en dimension 2 : la 1-périodicité, c'est-à-dire lorsque le treillis des vecteurs de périodicité est de dimension 1. Les ensembles de 1-périodes des SFTs de dimension 2 sont caractérisés à l'aide de la classe de complexité en espace **NSPACE**(2^n). En dimensions supérieures, la 1 périodicité ne peut plus être caractérisée en termes de classes de complexité et pourrait l'être en termes de calculabilité. Ensuite, nous nous intéressons aux ensembles de périodes des sous-shifts effectifs et sofiqes, donnant une caractérisation à l'aide d'ensembles Π_1^0 . Les limites de ce chapitre résident dans le fait que l'on ne parvient pas à des caractérisations de tous les types de treillis périodiques, on souhaiterait en particulier pouvoir caractériser, en dimension d , les treillis de dimension d de n'importe quelle forme.

Enfin, au chapitre 4, nous nous intéressons aux problèmes de la factorisation et de la conjugaison du point de vue de leur complexité calculatoire, prouvant que la conjugaison est Π_1^0 -complète pour les SFTs sur \mathbb{Z}^d pour $d \geq 2$ et Σ_3^0 -complète pour les sous-shifts effectifs de dimension $d \geq 1$ et sofiqes de di-

mension $d \geq 2$. Nous prouvons également que la factorisation est Σ_3^0 -complète pour les sous-shifts de type fini, sofiques et effectifs. Il est intéressant de noter que la conjugaison en dimensions $d \geq 2$ est au premier niveau de la hiérarchie arithmétique, et donc relativement simple : de la même complexité que décider si un SFT est vide. Le fait que la conjugaison soit plus simple que la factorisation pour les SFTs mais pas pour les effectifs et les sofiques est tout aussi intéressant.

Le graphe ci-dessous résume les dépendances entre chapitres :



La seule dépendance notable est que l'on utilise au chapitre 4 une construction que l'on introduit au chapitre 2 en section 2.2.1. On pourra d'ailleurs lire la section 2.2.1 seule avant de lire le chapitre 4, car elle est indépendante du reste du chapitre 2.

Chapitre 1

Définitions préliminaires

Sommaire

1.1	Sous-shifts	3
1.1.1	Définitions de base	3
1.1.2	SFTs et tuiles de Wang	6
1.1.3	Quelques opérations sur les sous-shifts	8
1.1.4	Aspects topologiques des sous-shifts	9
	La topologie usuelle	9
	Fonctions continues commutant avec le shift	10
1.1.5	Autres types de sous-shifts	11
	Sous-shifts sofiques	11
	Sous-shifts effectif	13
1.1.6	Sous-shifts de dimension 1	15
1.1.7	Périodicité, quasipériodicité et minimalité	17
1.1.8	Conjugaison et invariants de conjugaison	19
1.2	Calculabilité	22
1.2.1	Réductions et degrés Turing	23
1.2.2	Hierarchie arithmétique	24
1.2.3	Indécidabilité de la pavabilité du plan	25
	Liens entre SFTs et machines de Turing	25
	Jeu de tuiles de Robinson	29

Il existe deux formalismes permettant de décrire les objets que nous étudions ici : les pavages et les sous-shifts.

La formalisation en tant que pavages remonte à Wang [Wan61] et prend son origine dans l'étude de certains fragments de la logique du premier ordre. Une tuile de Wang est un carré unité dont les quatre bords sont colorés qui ne peut pas être tourné. Un jeu de tuiles est un ensemble fini de tuiles. On peut placer deux tuiles côte à côte uniquement si les bords se touchant sont de la même couleur. La question de la pavabilité du plan est de savoir si étant donné un jeu de tuiles τ , on peut placer des tuiles de τ à toutes les coordonnées du plan \mathbb{Z}^2 sans qu'il n'y ait d'erreur.

C'est dans ce formalisme qu'ont été prouvés la plupart des résultats liés à la décidabilité ou à la complexité calculatoire. Le résultat central, dû à Berger

[Ber64], est l'indécidabilité de la pavabilité du plan : il n'existe pas d'algorithme qui prenant en entrée un jeu de tuiles renvoie si oui ou non il pave le plan.

La formalisation en tant que sous-shifts a été introduite par Morse et Hedlund [MH38] afin d'étudier des systèmes dynamiques : étant donné un espace compact X et une transformation réversible ϕ , une trajectoire est une séquence de points de X obtenus par applications successives de ϕ :

$$\dots, \phi^{-2}(x), \phi^{-1}(x), x, \phi(x), \phi^2(x), \dots$$

Si l'on divise l'espace X en un nombre fini de partitions, chacune représentée par un symbole de $P = \{1, \dots, n\}$, on peut alors associer à chaque trajectoire un mot biinfini : à chaque point de la trajectoire on associe le symbole représentant la partition dans laquelle il se trouve. Décaler le mot à gauche correspond donc à appliquer ϕ à tous les points de la trajectoire, le décalé du mot correspond donc encore à une trajectoire. Les ensembles de mots ainsi créés sont appelés sous-shifts, car ils sont invariants par décalage. La figure 1.1 montre comment un espace X peut être partitionné et comment les trajectoires sont transformées en mots biinfinis.

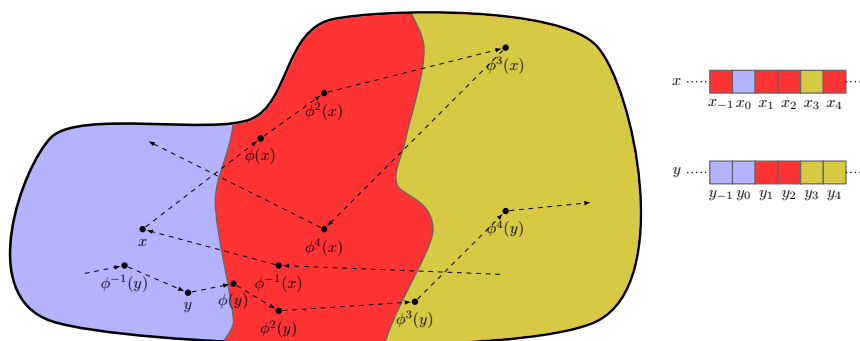


FIGURE 1.1 – Un système dynamique que l'on a divisé en trois partitions. On peut voir comment on a fait correspondre des mots biinfinis à deux trajectoires.

Ces mots biinfinis sont des éléments de $P^{\mathbb{Z}}$, mais dans la suite nous allons plutôt nous intéresser aux éléments de $P^{\mathbb{Z}^d}$, c'est à dire aux systèmes dynamiques ayant d transformations qui commutent, on appelle d la dimension du système. On verra dans ce chapitre que ces ensembles sont exactement ceux définis par des familles de mots interdits et nous nous intéresserons au cas particulier où cette famille est finie, c'est à dire aux sous-shifts de type fini (SFTs), qui, en dimension 2, correspondent aux pavages. Plus précisément, tout ensemble de pavages généré par un jeu de tuiles est conjugué à un SFT et inversement.

Pour une introduction plus complète à la dynamique symbolique, on pourra consulter l'excellent ouvrage de Lind et Marcus [LM95].

Dans ce chapitre, nous commençons par définir les sous-shifts en section 1.1, puis nous introduisons la classes des sous-shifts de type fini, et expliquons le lien avec les tuiles de Wang. Nous définirons également une topologie dans laquelle les sous-shifts sont des espaces compacts, permettant de ce fait d'utiliser des techniques d'extraction. Nous finirons cette section en introduisant les invariants de conjugaison. Ensuite, en section 1.2 nous effectuerons quelques rappels de

calculabilité et donnerons la preuve de l'indécidabilité de la pavabilité du plan de Robinson, dont nous réutiliserons la construction dans les chapitres 2 et 3.

1.1 Sous-shifts

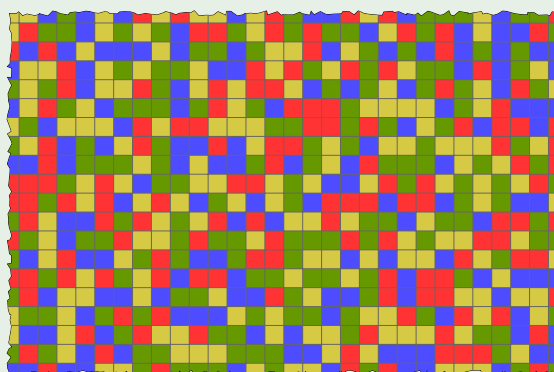
1.1.1 Définitions de base

Un ensemble fini de *symboles* Σ est appelé un *alphabet*. L'*espace des configurations* sur Σ en dimension d est l'ensemble des fonctions de \mathbb{Z}^d dans Σ , noté $\Sigma^{\mathbb{Z}^d}$. On appelle également l'espace des configurations le *full-shift*, pour une raison que nous donnerons ultérieurement.

Définition 1.1.1

Configuration

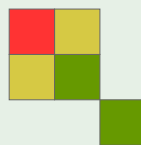
Une *configuration* est une fonction $c : \mathbb{Z}^d \rightarrow \Sigma$. Si on regarde Σ comme un ensemble de couleurs, c est alors un coloriage de \mathbb{Z}^d avec ces couleurs. Ci-dessous, on peut voir un extrait d'une configuration sur un alphabet à quatre symboles en dimension 2.



Nous allons en fait nous intéresser aux sous-ensembles du full-shift qui satisfont un ensemble de contraintes particulières : on souhaite garder uniquement des configurations où certains motifs n'apparaissent jamais.

Définition 1.1.2*Motif*

Un *motif* est une fonction $m : P \rightarrow \Sigma$, où $P \subset \mathbb{Z}^d$ est *fini* et appelé le *support* de m . m est donc un coloriage d'une portion finie de \mathbb{Z}^d . On notera P_m le support d'un motif m .

**Définition 1.1.3***Sous-shift*

Étant donné un alphabet Σ et une famille \mathcal{F} de motifs interdits, on appelle *sous-shift* l'ensemble $\mathcal{X}_{\mathcal{F}}$ des configurations ne contenant aucun des motifs de \mathcal{F} . C'est à dire

$$\mathcal{X}_{\mathcal{F}} = \left\{ c \in \Sigma^{\mathbb{Z}^d} \mid \forall z \in \mathbb{Z}^d, \forall m \in \mathcal{F}, c|_{z+P_m} \neq m \right\}.$$

Les configurations de $\mathcal{X}_{\mathcal{F}}$ sont appelées les *points* du sous-shift, et d est la *dimension* du sous-shift. Dans la suite, on notera Σ_X l'alphabet d'un sous-shift X . Par abus de notation, on notera parfois simplement \mathcal{F} le couple normalement formé de \mathcal{F} et Σ .

La figure 1.2 montre un exemple de sous-shift, sur deux symboles \blacksquare et \square , défini par une famille finie de motifs interdits \mathcal{F} .

Étant donné une famille de motifs interdits \mathcal{F} , un motif sera dit *admissible* s'il ne contient aucun motif de \mathcal{F} . Un motif sera appelé *extensible* s'il apparaît dans un point du sous-shift $\mathcal{X}_{\mathcal{F}}$. Il est important de noter que si un motif extensible est nécessairement admissible, l'inverse n'est pas nécessairement vrai.

Pour une famille de motifs interdits \mathcal{F} , on notera $\mathcal{A}_n(\mathcal{F})$ (resp. $\mathcal{E}_n(\mathcal{F})$) l'ensemble des motifs admissibles (resp. extensibles) de support \mathcal{B}_n , la boule $\llbracket -n, n \rrbracket^d$ de rayon n et de diamètre $2n + 1$. On appelle un motif de support \mathcal{B}_n un *bloc* de taille n ou encore *n-bloc*.

Un même sous-shift peut être défini par de nombreuses familles de motifs interdits différentes. Notons que la notion de motif admissible est liée à l'ensemble de motifs interdits considéré, tandis que la notion de motif extensible est uniquement liée au sous-shift.

Le nom sous-shift vient du fait que ces espaces sont invariants par translation (ou *shift-invariants*), également appelée *shift* en anglais. Pour $\mathbf{z} \in \mathbb{Z}^d$, on note $\sigma_{\mathbf{z}}$ l'opérateur de translation par \mathbf{z} . Si x est une configuration, alors σ agit sur x par

$$\forall \mathbf{p} \in \mathbb{Z}^d, \sigma_{\mathbf{z}}(x)(\mathbf{p}) = x(\mathbf{z} + \mathbf{p}).$$

Un espace X tel que $\sigma_{\mathbf{z}}(X) = X$ pour tout $\mathbf{z} \in \mathbb{Z}^d$ est appelé invariant par translation, ou encore *shift-invariant*. Les espaces invariants par translation ne

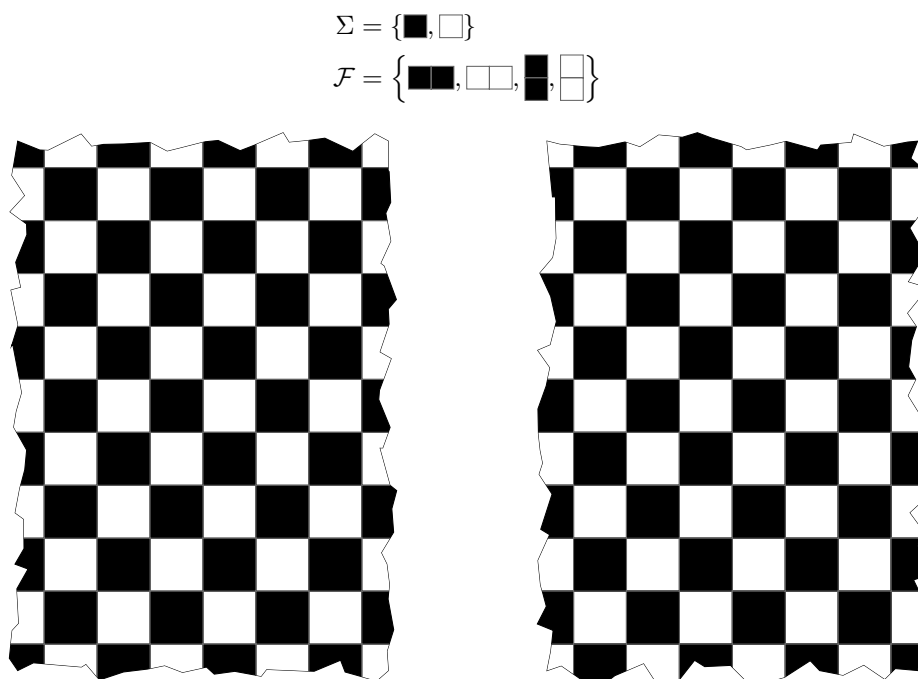


FIGURE 1.2 – Le sous-shift $\mathcal{X}_{\mathcal{F}}$ ne contient que deux configurations différentes, l'une ayant le symbole \blacksquare à l'origine, l'autre le symbole \square .

sont pas nécessairement des sous-shifts.

1.1.2 SFTs et tuiles de Wang

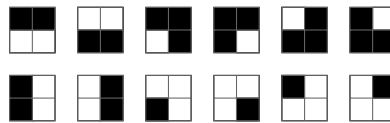
Notons que dans la section précédente, nous n'avons imposé aucune restriction sur la famille de motifs interdits \mathcal{F} . En fonction du type de famille que l'on autorise, on définit des classes de sous-shifts différentes. La plus naturelle est celle des sous-shifts de type fini, qui sont le principal objet d'étude de cette thèse.

Définition 1.1.4

Sous-shift de type fini (SFT)

Un *sous-shift de type fini* est un sous-shift X qui peut être défini par une famille de motifs interdits \mathcal{F} finie.

Il est utile de voir que l'on peut aisément convertir une famille finie de motifs interdits en une autre définissant le même sous-shift mais dont les motifs ont tous le même support. Par exemple, le sous-shift de la figure 1.2 peut également être défini par des familles de motifs dont les supports sont tous \mathcal{B}_n pour n'importe quel $n \geq 2$, pour $n = 2$ on a les motifs suivants :



On appelle *rayon* d'une famille de motifs interdits \mathcal{F} , le plus petit r tel que les supports de tous les motifs de \mathcal{F} soient inclus dans \mathcal{B}_r . Le rayon d'un SFT X peut alors être défini comme le plus petit r tel qu'il y ait une famille \mathcal{F} de rayon r telle que $X = \mathcal{X}_{\mathcal{F}}$.

Un SFT peut être défini aussi bien par motifs interdits que par *motifs autorisés* : pour une famille de motifs interdits \mathcal{F} de rayon r les motifs autorisés sont $\Sigma^{\mathcal{B}_r} \setminus \mathcal{F}$, le complémentaire des motifs interdits sur leur domaine.

La classe des SFTs est à mettre en relation avec les tuiles de Wang [Wan61].

Définition 1.1.5*Tuile de Wang*

Soit Q un ensemble fini de couleurs, une *Tuile de Wang* est un quadruplet d'éléments de Q . On peut voir une tuile de Wang comme un carré dont les côtés sont coloriés :



Un *jeu de tuiles* est un ensemble fini de tuiles de Wang. On dit qu'un jeu de tuiles τ pave le plan si l'on peut placer sur chaque point de \mathbb{Z}^2 une copie d'une des tuiles de τ de manière à ce que deux tuiles adjacentes aient leurs bords communs de même couleur. On appelle une telle décoration de \mathbb{Z}^2 un *pavage valide* ou simplement *pavage*. L'ensemble des pavages valides par un jeu de tuiles est un *espace de pavages*.

La figure 1.3 montre un exemple de jeu de tuiles ainsi qu'un pavage associé.

$$\tau = \{\square, \square\}$$

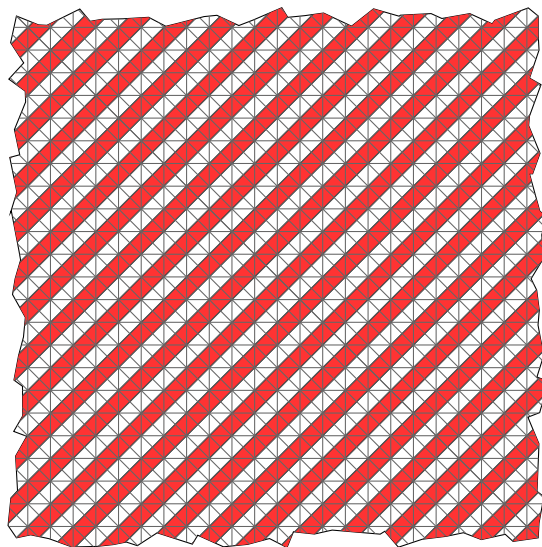


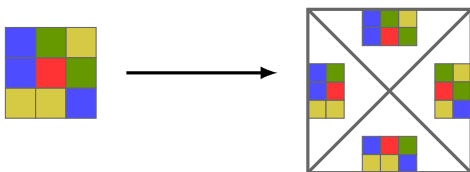
FIGURE 1.3 – Un exemple de jeu de tuiles et un pavage associé.

Notons que les pavages par le jeu de tuiles τ de la figure 1.3 et les points du sous-shift de la figure 1.2, sont égaux à renommage des tuiles/symboles près. Mais est-ce que tout SFT peut être “converti” en espace de pavages et inversement ?

Il est clair qu'un espace de pavages est un SFT : on peut voir τ comme un alphabet, la famille de motifs interdits est alors formée de tous les motifs 2×1

et 1×2 dont les couleurs sur le côté adjacent ne sont pas les mêmes. L'ensemble des pavages est donc un sous-shift de $\tau^{\mathbb{Z}^2}$. On notera \mathcal{X}_τ le SFT de dimension 2 produit par un jeu de tuiles.

Réciproquement, une famille de motifs interdits \mathcal{F} peut être convertie en jeu de tuiles de manière à ce que son espace de pavages soit “isomorphe¹” au SFT $\mathcal{X}_\mathcal{F}$: On peut d’abord supposer, sans perte de généralité, que le support des motifs interdits de \mathcal{F} est \mathcal{B}_n . À chaque motif autorisé par \mathcal{F} on associe la tuile de Wang où la couleur du haut est formée des $n - 1$ lignes du haut du motif, celle du bas des $n - 1$ lignes du bas, celle de gauche des $n - 1$ colonnes de gauche et celle de droite des $n - 1$ colonnes de droite. La figure ci-dessous montre comment à partir d’un motif 3×3 autorisé, on construit une tuile de Wang.



Un pavage par ce jeu de tuiles correspond exactement à une configuration de $\mathcal{X}_\mathcal{F}$, et on peut déduire de chaque tuile le symbole associé dans l’alphabet du SFT.

Ce lien entre les SFTs et tuiles de Wang est important, dans le sens où la plupart des résultats ayant trait à l’aspect algorithmique ont été prouvés dans ce formalisme. Par exemple, Berger [Ber64, Ber66] a prouvé qu’il était indécidable de savoir si un jeu de tuiles donné pouvait paver le plan. C’est à dire qu’il est indécidable de savoir si un SFT est vide. Levin [Lev86] a par exemple utilisé les tuiles de Wang pour donner un problème complet lorsqu’il a défini la complexité moyenne d’un problème. Harel [Har85] s’en est servi afin de montrer des problèmes complets pour certaines classes de complexité, mais aussi pour certaines classes d’indécidabilité. Emde Boas [Emd97] a d’ailleurs montré que les tuiles de Wang sont un outil idéal pour faire des réductions.

1.1.3 Quelques opérations sur les sous-shifts

On présente ici quelques opérations sur les sous-shifts de même dimension par lesquelles la classe² du sous-shift transformé est stable :

- La première opération est l’*union disjointe* que l’on note \sqcup . Si X, Y sont deux sous-shifts du même type, alors leur union disjointe $X \sqcup Y$ est le sous-shift défini sur l’alphabet $\Sigma_X \sqcup \Sigma_Y$ où les règles sur chaque alphabet sont conservées et où l’on interdit en plus à des symboles de Σ_X d’être voisins de symboles de Σ_Y .
- La seconde est la *superposition*, que l’on peut également voir comme un produit cartésien : Soient X, Y deux sous-shifts du même type, le produit cartésien $X \times Y$, peut être vu en pratique comme un sous-shift défini sur l’alphabet $\Sigma_X \times \Sigma_Y$ où les règles de X (resp. Y) s’appliquent sur la

1. La bonne notion est la conjugaison, nous la définissons plus loin.

2. Nous verrons dans la suite d’autres classes de sous-shifts, comme les sous-shifts effectifs ou sofiques, qui sont également stables par ces opérations.

composante des symboles de Σ_X (resp. Σ_Y). On peut également interdire certaines superpositions de symboles en retirant simplement celles-ci de l'alphabet ou encore en rajoutant cette superposition comme motif interdit. En pratique, même lorsque l'on imposera des conditions de superpositions, on notera toujours (de manière abusive) la superposition $X \times Y$, bien que formellement ce soit un sous-shift inclus dans celle-ci. On appellera X et Y des *couches*, ou encore des *composantes* du sous-shift.

1.1.4 Aspects topologiques des sous-shifts

La topologie joue un rôle important dans l'étude des sous-shifts, en grande partie car ceux-ci sont des espaces compacts et que les techniques d'extraction y sont donc utilisables. On pourra consulter le livre de Kuratowski [Kur66] pour des rappels de topologie.

La topologie usuelle

On peut définir une topologie sur les configurations à l'aide de la distance

$$d(x, y) = 2^{-\min\{\|z\|_\infty \mid z \in \mathbb{Z}^d, x(z) \neq y(z)\}}$$

où $\|z\|_\infty = \max_i |z_i|$. Avec cette distance, plus la portion sur laquelle deux configurations coïncident au centre est grande, plus elles sont proches. Dans cette topologie, les cylindres

$$C_m = \left\{ x \in \Sigma^{\mathbb{Z}^d} \mid x|_{P_m} = m \right\},$$

où m est un motif, forment une base d'ouverts/fermés. Cette topologie s'étend naturellement aux sous-shifts, qui sont des espaces compacts dans celle-ci :

Théorème 1.1.1 (Compacité des $\mathcal{X}_{\mathcal{F}}$). *Pour toute famille de motifs interdits \mathcal{F} , l'espace $\mathcal{X}_{\mathcal{F}}$ est compact.*

Preuve. Soit $(c_i^0)_{i \in \mathbb{N}}$ une suite de configurations de $\mathcal{X}_{\mathcal{F}}$. D'après le principe Pigeon-Trou³, il existe une infinité de configurations de (c_i^0) contenant le même symbole au centre. On note $(c_i^1)_{i \in \mathbb{N}}$ cette suite de configurations. Parmi les $(c_i^1)_{i \in \mathbb{N}}$ il y a une infinité de configurations coïncidant sur la boule de rayon 2. On construit ainsi une infinité de suites $(c_i^n)_{i \in \mathbb{N}}$, où (c_i^{n+1}) est extraite de (c_i^n) et dont tous les éléments sont identiques sur la boule de rayon $n + 1$ au centre. Si l'on prend la sous-suite $(c_i^i)_{i \in \mathbb{N}}$ de (c_i^0) celle-ci converge vers la configuration c qui est égale à c_0^n sur la boule de rayon n . Comme toute zone arbitrairement grande de c apparaît dans une configuration valide, $c \in \mathcal{X}_{\mathcal{F}}$. \square

Une conséquence intéressante de la compacité des sous-shifts, est que si l'on prend un sous-shift $X = \mathcal{X}_{\mathcal{F}}$, et que $(x_i)_{i \in \mathbb{N}}$ est une suite de configurations pour laquelle le i -bloc au centre de x_i est *admissible*, alors $x_i \rightarrow x \in X$.

La condition de shift-invariance à elle seule ne suffit pas à caractériser les sous-shifts, elle n'implique en effet pas la compacité de l'espace : par exemple

³. Certaines personnes appellent ce principe, *principe de Dirichlet*, *principe Chaussette-Tiroir*, ou encore simplement *lemme des tiroirs*, l'auteur de ce mémoire pense, éventuellement à tort, qu'ils ont tort et que l'appellation *principe Pigeon-Trou* mériterait d'être généralisée.

l'ensemble des configurations contenant *exactement* un symbole 1, est shift-invariant mais n'est pas un sous-shift, car la suite c_i de configurations où le 1 est en position (i, i) converge vers la configuration uniforme avec uniquement des 0 ne faisant pas partie de l'ensemble.

Théorème 1.1.2. *Les sous-shifts sont exactement les sous-ensembles fermés, shift-invariants de $\Sigma^{\mathbb{Z}^d}$.*

Preuve. Soit $X \subseteq \Sigma^{\mathbb{Z}^d}$ un ensemble fermé, shift-invariant. Soit F l'ensemble des motifs n'apparaissant dans aucune configuration de X . Clairement $X \subseteq \mathcal{X}_F$. Prenons maintenant une configuration $c \in \mathcal{X}_F$, par shift-invariance, tout motif de c apparaît au centre d'une configuration de X , on a donc une suite de configurations de X convergeant vers c . Comme X est fermé, $c \in X$, donc $X \subseteq \mathcal{X}_F$.

Soit \mathcal{X}_F un sous-shift. Il est shift invariant et d'après le théorème 1.1.1 il est fermé. \square

Fonctions continues commutant avec le shift

Après avoir défini une topologie, il est naturel de considérer les fonctions continues par rapport à cette topologie. Cependant, dans notre cadre les fonction intéressantes sont celles qui commutent avec le shift. On dit qu'une application g commute avec le shift si $g \circ \sigma_z = \sigma_z \circ g$ pour tout $z \in \mathbb{Z}^d$. En particulier, il suffit que g commute avec les σ_{e_i} où les e_i forment la base usuelle de \mathbb{Z}^d .

Théorème 1.1.3 (Curtis, Hedlund, Lyndon [Hed69]). *Soient X, Y deux sous-shift quelconques et $G : X \rightarrow Y$ une application continue commutant avec le shift, alors il existe r et $g : \Sigma_X^{B_r} \rightarrow \Sigma_Y$ tels que*

$$\forall x \in X, \forall z \in \mathbb{Z}^d, G(x)(z) = g(x|_{z+B_r}).$$

On appellera r le rayon de g .

Preuve. Application directe de la définition de continuité. \square

Ce théorème montre, en substance, que si une application continue commute avec le shift, alors elle est locale. Par abus de notation on notera souvent de la même manière la fonction locale g et la fonction globale G .

Les fonctions de ce type le plus souvent étudiées dans notre cadre sont les fonctions de conjugaison et de factorisation. Nous les étudierons plus en détail par la suite.

Définition 1.1.6

Conjugaison

Soient X, Y deux sous-shifts, on dit qu'ils sont *conjugués* si et seulement s'il existe une *fonction de conjugaison* de l'un vers l'autre : une fonction continue *bijective* commutant avec le shift $F : X \rightarrow Y$.

Définition 1.1.7*Factorisation*

Soient X, Y deux sous-shifts, on dit que X se *factorise* sur Y si et seulement s'il existe une fonction continue *surjective* commutant avec le shift $F : X \rightarrow Y$. On dit que Y est un *facteur* de X .

Définition 1.1.8*Plongement*

Soient X, Y deux sous-shifts, on dit que X se *plonge* dans Y , on note $X \hookrightarrow Y$, si et seulement s'il existe une fonction continue *injective* commutant avec le shift $F : X \rightarrow Y$. On dit que F est un *plongement* de X dans Y .

1.1.5 Autres types de sous-shifts

Nous avons mentionné précédemment l'existence d'autres types de sous-shifts : les sous-shifts sofiques et effectifs. Ceux-ci sont des classes de sous-shifts qui recouvrent respectivement les facteurs de SFTs et les sous-shifts dont les contraintes sont descriptibles algorithmiquement.

Sous-shifts sofiques

Une extension naturelle des tuiles de Wang est de rajouter une couleur supplémentaire au centre de chaque tuile, comme en figure 1.4a. On peut alors pour chaque pavage, ne conserver que cette information par le biais d'une projection. Ainsi si la couleur supplémentaire vient d'un alphabet C , chaque pavage est alors associé à un coloriage du plan par C , donc une configuration. On appellera ces tuiles de Wang des tuiles décorées, et l'espace de pavages associé un espace de pavages décoré.

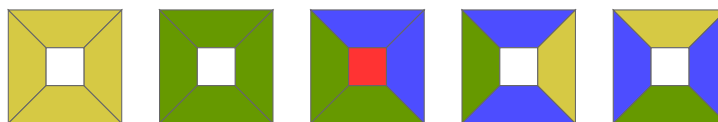
Les ensembles de configurations générés de cette manière sont appelés des sous-shifts sofiques. La classe des sous-shifts sofiques a été introduite initialement par Weiss [Wei73] comme la clôture par factorisation de la classe des SFTs.

Définition 1.1.9*Sous-shift sofique*

Un sous-shift Y est *sofique* si et seulement s'il existe un SFT X une fonction continue *surjective* commutant avec le shift $G : X \rightarrow Y$ telle que $G(X) = Y$

Il faut toutefois prouver que les espaces ainsi définis sont bien des sous-shifts : **Preuve.** X étant compact et G étant continue, $G(X)$ est compact. De plus, $G(X)$ est shift-invariant. \square

Il y a au moins trois définitions équivalentes des sous-shifts sofiques, dont les deux précédentes :



(a) Des tuiles de Wang avec une information supplémentaire au centre.

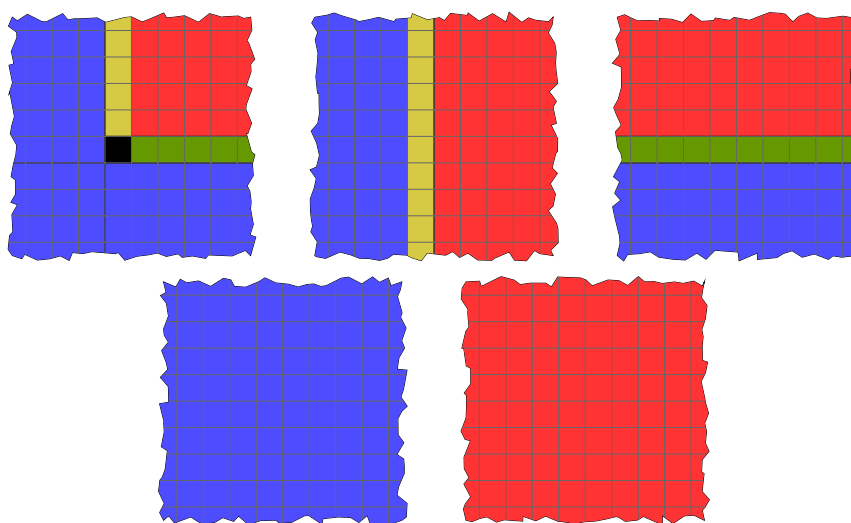
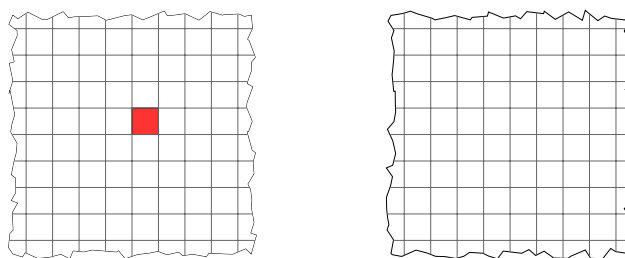
(b) Les configurations du sous shift X , il peut être défini par un ensemble fini de contraintes.(c) Les configurations du sous shift Y , qui ne peut pas être défini à l'aide d'une collection finie de motifs interdits.

FIGURE 1.4 – En (b) et (c), deux sous-shifts X, Y , X se factorise sur Y en transformant le symbole \blacksquare en \blacksquare , et en transformant tous les autres symboles en \square . En (a) des tuiles de Wang avec lesquelles tout pavage valide se transforme en une configuration de Y quand on ne retient que la couleur au centre de chaque tuile. Notons que le sous-shift Y n'est pas de type fini.

1. Les sous-shifts sofiques sont les facteurs de SFTs.
2. Les sous-shifts sofiques sont les projections des espaces de pavages décorés sur leur décoration.
3. Les sous-shifts sofiques sont les projections vers un alphabet C_2 de SFTs définis sur un alphabet $C_1 \times C_2$.

Preuve. On va prouver l'équivalence de ces définitions dans le sens (1) \Rightarrow (3) \Rightarrow (2) \Rightarrow (1) :

- (1) \Rightarrow (3) Soit Y un sous-shift sofique défini par un SFT X et une factorisation G de rayon r . Prenons le SFT $X \times Y$ où un symbole a ne peut apparaître sur la deuxième composante Y que si c'est l'image par G de son voisinage \mathcal{B}_r . Clairement, les configurations formées uniquement de la deuxième composante sont exactement les configurations de Y .
- (3) \Rightarrow (2) Soit Y un sous-shift sofique défini par un sous-shift $\mathcal{X}_{\mathcal{F}} \times Y$ et la projection lettre à lettre canonique π , on a vu dans la section 1.1.2 comment on pouvait transformer un SFT en tuiles de Wang. On transforme donc \mathcal{F} en tuiles de Wang, que l'on décore avec la lettre de Y associée.
- (2) \Rightarrow (1) Les tuiles de Wang décorées forment l'alphabet du SFT constitué par ses pavages, la fonction de factorisation est alors triviale. □

La classe des sous-shifts sofiques contient les SFTs. Mais un sous-shift sofique n'est pas nécessairement de type fini. Par exemple le sous-shift Y défini à la figure 1.4c n'est pas de type fini, alors qu'il est facteur de X qui est de type fini :

Preuve. Si Y était de type fini, alors il existerait une famille \mathcal{F} de motifs interdits de rayon r telle que $Y = \mathcal{X}_{\mathcal{F}}$. La configuration c_1 contenant exactement un point noir et la configuration c_2 en contenant exactement deux à distance plus grande que $3r$ ayant exactement les mêmes motifs de taille r appartiendraient alors toutes les deux à $\mathcal{X}_{\mathcal{F}}$. Donc Y n'est pas un SFT. □

Sous-shifts effectif

Les SFTs et sous-shifts sofiques sont intéressants parce qu'ils sont simples à décrire, c'est à dire qu'ils sont définissables de manière finie. Une autre classe de sous-shifts intéressante est celle de sous-shifts effectifs, c'est à dire ceux que l'on peut définir à l'aide d'une famille de motifs interdits générée par un algorithme.

Définition 1.1.10

Sous-shift effectif

Un *sous-shift effectif* X est un sous-shift pour lequel il existe une famille de motifs interdits récursivement énumérable \mathcal{F} telle que $X = \mathcal{X}_{\mathcal{F}}$. C'est à dire qu'il existe une machine de Turing M qui énumère les motifs de \mathcal{F} .

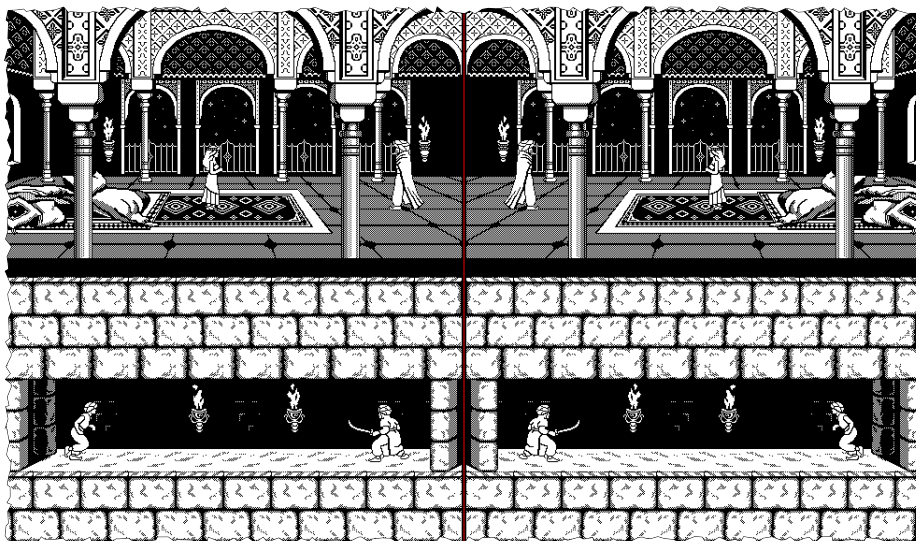
On a en figure 1.5 un exemple de sous-shift effectif, celui-ci est constitué de :

- toutes les configurations sur l'alphabet $\{\blacksquare, \square\}$
- dès que le symbole \blacksquare apparaît, alors il n'y a que ce symbole dans la même colonne, et aucune autre colonne avec celui-ci. De plus, la partie à gauche de la colonne est la symétrique de celle à droite.

$$\Sigma = \{\text{red}, \text{black}, \text{white}\}$$

$$\mathcal{F} = \left\{ \begin{array}{c} \text{black} \\ \text{red} \end{array}, \begin{array}{c} \text{white} \\ \text{red} \end{array}, \begin{array}{c} \text{red} \\ \text{black} \end{array}, \begin{array}{c} \text{red} \\ \text{white} \end{array} \right\} \cup \bigcup_{n \in \mathbb{N}} \left\{ \begin{array}{c} \text{red} \quad \text{red} \quad \text{white} \quad \text{red} \quad \text{black} \quad \text{black} \quad \text{red} \quad \text{white} \\ \underbrace{\hspace{1cm}}_n \quad \underbrace{\hspace{1cm}}_n \quad \underbrace{\hspace{1cm}}_n \quad \underbrace{\hspace{1cm}}_n \quad \underbrace{\hspace{1cm}}_n \end{array} \right\}$$

(a) Les quatre premiers motifs forcent un colonne à être composée soit uniquement de rouges, soit de blancs et/ou de noirs. Les motifs suivants forcent à ce qu'il y ait au plus une colonne rouge et à ce que quand une colonne rouge apparaît les symboles soient symétriques des deux côtés.



(b) Un point de ce sous-shift comportant une colonne rouge, la partie à gauche de la colonne est la symétrique par rapport à celle-ci de la partie à droite.

FIGURE 1.5 – Exemple de sous-shift effectif : en (a) les motifs interdits qui le définissent et en (b), un exemple de configuration comportant une ligne rouge. Les configurations ne comportant pas de ligne rouge sont toutes les configurations sur les deux symboles restant.

On peut aisément remarquer qu'un sous-shift est effectif s'il existe une famille récursive de motifs interdits le définissant. En effet, on peut transformer une famille de motifs interdits récursivement énumérable en famille de motifs interdits récursive en rendant son énumération croissante : l'énumération fait croître le support des motifs. Pour faire cela, on prend une énumération quelconque, à chaque nouveau motif interdit m de l'énumération initiale, on fait correspondre un nombre fini de motifs interdits de support \mathcal{B}_n , avec $P_m \subseteq \mathcal{B}_n$, qui sont tous les motifs m' de $\Sigma^{\mathcal{B}_n}$ tels que $m'|_{P_m} = m$. On a ainsi rendu l'énumération des motifs interdits croissante, on peut donc décider si un motif appartient à notre nouvel ensemble de motifs interdits, ce dernier est donc récursif.

Un sous-shift sofique est effectif, en revanche un sous-shift effectif n'est pas nécessairement sofique : Par exemple, le sous-shift de la figure 1.5, que l'on notera X , n'est pas sofique :

Preuve. Supposons que X soit sofique, il existe alors un jeu de tuiles de Wang

décorées τ contenant K tuiles et une projection de rayon 1 π telle que $\pi(\mathcal{X}_\tau) = X$. On s'intéresse maintenant uniquement aux configurations contenant une ligne rouge verticale d'abscisse 0 : le symbole en (i, j) est le même que celui en $(-i, j)$.

Prenons une boule $B = z + \mathcal{B}_n$ avec $z \in \mathbb{Z}^d$, telle que pour tout i , $(0, i) \notin B$: il y a 2^{n^2} motifs possibles avec ce support dans ces configurations. Il y a donc au moins 2^{n^2} préimages distinctes de ces motifs par π qui apparaissent dans celles-ci. Mais il y a au plus K^{4n} bordures possibles pour ces préimages, donc si l'on a choisi n suffisamment grand, il y a deux préimages m_1, m_2 de motifs différents ayant la même bordure⁴. Soit x_1 une configuration qui contient l'image de m_1 et x_2 une qui contient l'image de m_2 , on peut remplacer m_1 dans la préimage de x_1 par m_2 , l'application de π donne alors une configuration contenant une ligne rouge mais non symétrique par rapport à celle-ci. X n'est donc pas sofique. \square

On peut adapter cette preuve en n'importe quelle dimension, prouvant ainsi que les sous-shifts effectifs de dimension d ne sont pas des sous-shifts soifiques de dimension d . Il existe cependant un lien entre les sous-shifts soifiques de dimension $d+1$ et les sous-shifts effectifs de dimension d . On peut prouver que le relèvement d'un sous-shift effectif en dimension supérieure est sofique.

Définition 1.1.11

Relèvement

Soit $X \subseteq \Sigma^{\mathbb{Z}^d}$ un sous-shift de dimension d , le relèvement de X en dimension $d+k$ est le sous-shift $X' \subseteq \Sigma^{\mathbb{Z}^{d+k}}$ dont les configurations sont les configurations de X identiques sur les dimensions supplémentaires.

Par exemple, le relèvement d'un sous-shift X de dimension 1 en dimension 2 est le sous-shift constitué des configurations dont toutes les lignes sont identiques et égales à un point de X .

Théorème 1.1.4 (AS, Hochman [Hoc09b], Durand, Romashchenko et Shen [DRS10]). *Un sous-shift X de dimension d est effectif si et seulement si son relèvement en dimension $d+1$ est sofique.*

Ce théorème nous permettra en particulier de construire des sous-shifts soifiques de dimension $d+1$ simplement en décrivant le sous-shift effectif de dimension d correspondant.

1.1.6 Sous-shifts de dimension 1

Bien que les objets que nous étudions principalement dans cette thèse soient des sous-shifts de dimension supérieure à 2, nous aurons occasionnellement besoin de résultats sur les sous-shifts de dimension 1. Les SFTs et sous-shifts soifiques de dimension 1 ont été extensivement étudié, un excellent ouvrage récapitulant les principaux résultats est le livre de Lind et Marcus [LM95]. Nous

4. Il s'agit à nouveau d'une application du principe Pigeon-Trou.

nous intéressons ici spécifiquement à un aspect, celui de la périodicité :

Définition 1.1.12

Un point $x \in \Sigma^{\mathbb{Z}}$ est périodique si son orbite $\{\sigma_k(x) \mid k \in \mathbb{Z}\}$ est finie.

De manière équivalente un point x est périodique s'il existe un mot $w \in \Sigma^*$ tel que $x = \dots www \dots$. Nous aurons besoin de la propriété suivante par la suite :

Propriété 1.1.1. *En dimension 1 tout SFT non vide contient un point périodique.*

Afin de montrer cette propriété, nous allons montrer le lien qu'il peut y avoir entre les sous-shifts sofiques de dimension 1 et les graphes orientés. Un graphe est un couple (V, E) . Les éléments de V sont appelés noeuds et ceux de E des arêtes. Une arête a a un noeud de départ noté $t(a)$ et un noeud d'arrivée noté $h(a)$.

Définition 1.1.13

SFT engendré par un graphe

Soit $G = (V, E)$ un graphe orienté, on définit le sous-shift \mathcal{X}_G engendré par G de la manière suivante :

$$\mathcal{X}_G = \{x \in E^{\mathbb{Z}} \mid \text{Pour tout } i \in \mathbb{Z}, h(x_i) = t(x_{i+1})\}$$

Un tel sous-shift est clairement de type fini, les motifs interdits étant tous les mots fg tels que $h(f) \neq t(g)$. Ce que l'on souhaite montrer est que tout sous-shift de type fini est équivalent à un tel graphe. Il y a toutefois un problème : tel qu'exprimé ici, le SFT engendré par un graphe est nécessairement de rayon 1, alors qu'il y a des SFTs de rayons plus grands. On peut cependant transformer n'importe quel SFT de rayon r en un SFT de rayon 1 en augmentant simplement l'alphabet, qui est alors Σ^{2r+1} . C'est-à-dire en incorporant la mémoire dans l'alphabet. Ainsi le point

$$\dots x_{-r} \dots x_{-1} x_0 x_1 \dots x_r \dots$$

devient le point

$$\dots \begin{pmatrix} x_{-2r} \\ \vdots \\ x_{-r} \\ \vdots \\ x_0 \end{pmatrix} \dots \begin{pmatrix} x_{-r-1} \\ \vdots \\ x_{-1} \\ \vdots \\ x_{r-1} \end{pmatrix} \begin{pmatrix} x_{-r} \\ \vdots \\ x_0 \\ \vdots \\ x_r \end{pmatrix} \begin{pmatrix} x_{-r+1} \\ \vdots \\ x_1 \\ \vdots \\ x_{r+1} \end{pmatrix} \dots \begin{pmatrix} x_0 \\ \vdots \\ x_r \\ \vdots \\ x_{2r} \end{pmatrix} \dots$$

dans le nouveau SFT, qui est maintenant de rayon 1. On obtient donc le théorème suivant :

Théorème 1.1.5. *Tout SFT X de dimension 1 est représentable par un graphe G , dont les chemins biinfinis correspondent aux points de X .*

On peut directement déduire de ce théorème la propriété 1.1.1, car pour qu'il y ait une marche biinfinie sur un graphe, il est nécessaire que ce dernier contienne un cycle : un chemin passant uniquement par ce cycle est périodique et appartient au SFT.

La propriété 1.1.1 n'est en revanche pas nécessairement vraie pour les sous-shifts de dimension 1 qui ne sont pas de type fini ou sofiques. Par exemple le sous-shift effectif où l'on interdit tous les mots $awawa$, où a est une lettre et w un mot, ne peut pas contenir de point périodique mais est non-vide car il contient au moins le mot de Prouhet-Thue-Morse [Thu12, Mor21].

En dimension 2, une généralisation de cette représentation par un graphe a été donnée par Nasu [Nas95] sous la forme des *systèmes textiles* : deux graphes et un homomorphisme entre ces graphes.

1.1.7 Périodicité, quasipériodicité et minimalité

La périodicité est une propriété des configurations essentielle à l'étude des SFTs. On a vu, avec la propriété 1.1.1, qu'en dimension 1 tout SFT non-vide contenait un point périodique. Cette particularité permet de décider algorithmiquement si un SFT est vide, grâce au fait que si un SFT est vide, par compacité, il existe un rayon r tel qu'il n'existe aucun motif admissible de support \mathcal{B}_r . La procédure est alors simple, on cherche à remplir des boules \mathcal{B}_r pour r de plus en plus grand jusqu'à ce que cela soit impossible, ou que l'on remarque que l'on peut construire une configuration périodique.

Mais en dimensions supérieures, un SFT non vide n'a pas forcément de point ayant un vecteur de périodicité et savoir si un SFT est vide devient indécidable, comme nous le verrons plus tard (section 1.2.3).

Définition 1.1.14

Vecteur de périodicité

Soit c une configuration, on appelle vecteur de périodicité de c tout vecteur $\mathbf{v} \in \mathbb{Z}^d$ tel que $\sigma_{\mathbf{v}}(c) = c$, on dira aussi que c est périodique selon \mathbf{v} . Le vecteur nul est toujours un vecteur de périodicité.

Pour les SFTs de dimension $d = 1$, avoir un vecteur de périodicité non nul est équivalent à avoir une orbite $\{\sigma_{\mathbf{z}}(x) \mid \mathbf{z} \in \mathbb{Z}^d\}$ finie. On peut donc généraliser la notion de périodicité aux dimensions supérieures en gardant cette définition par orbites finies. Celle-ci est équivalente à la définition en termes de vecteurs de périodicité suivante :

Définition 1.1.15

Configuration périodique

Une configuration c est périodique s'il existe $n \in \mathbb{N}^*$ tel que pour tout $1 \leq i \leq d$, $\sigma_{ne_i}(c) = c$.

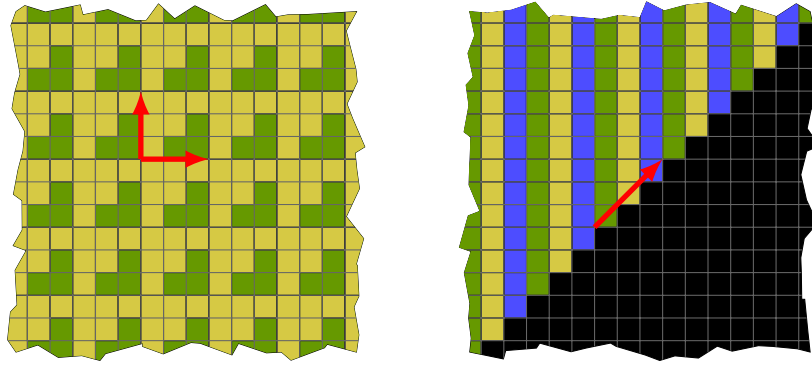


FIGURE 1.6 – A gauche un fragment de configuration périodique, les deux vecteurs de périodicité sont mis en évidence. A droite une configuration avec un vecteur de périodicité non-nul.

Cette définition est aussi équivalente à avoir d vecteurs de périodicité non colinéaires. Nous étudierons de manière approfondie les notions de périodicité en dimensions $d \geq 2$ au chapitre 3.

Wang [Wan60, Wan61], après avoir introduit les tuiles, a conjecturé que la pavabilité du plan (savoir si un SFT en dimension 2 est non-vide) était décidable. Cette conjecture, fautive, était partiellement due à l'intuition conférée par le théorème suivant, qui ramène la notion de périodicité à l'existence d'un vecteur de périodicité :

Théorème 1.1.6. *En dimension 2, tout SFT admettant un point ayant un vecteur de périodicité non nul admet un point périodique.*

Remarque 1.1.1. *Nous ne donnons pas la preuve de ce théorème ici, car elle découle directement du lemme 3.1.2 que nous montrerons plus tard et de la propriété 1.1.1.*

Ceci laissait penser que la situation allait être similaire à la dimension 1 en dimensions supérieures. Ce n'est pas le cas : il existe des SFTs non vides ne contenant aucun point admettant un vecteur de périodicité non nul. L'existence de tels SFTs, que l'on nomme *apériodiques* n'est pas évidente, Berger [Ber66, Ber64] montra en même temps leur existence et l'indécidabilité de la pavabilité du plan. Par la suite de nombreux SFTs apériodiques ont été construits par Robinson [Rob71], Kari [Kar96], Culik II [Cul96], Durand, Romashchenko et Shen [DRS09], Durand, Romashchenko et Shen [DRS12], Poupet [Pou10].

La figure 1.6 montre une configuration périodique et une configuration avec un seul vecteur de périodicité en dimension 2.

En dimension supérieure à 2, un SFT n'a pas nécessairement de point périodique, mais on peut néanmoins prouver que tous les SFTs ont au moins une configuration quasipériodique, et donc avec une certaine régularité :

Définition 1.1.16*Configuration quasipériodique/minimale*

Une configuration c est *quasipériodique/minimale* pour tout motif p de taille n , il existe m , tel que dans toute position $z \in \mathbb{Z}^d$, $c|_{z+\mathcal{B}_m}$ contient le motif p . On appelle \mathcal{B}_m la *fenêtre* de p . On dira qu'une configuration est *strictement quasipériodique* si elle est quasipériodique mais pas périodique.

Ainsi, une configuration périodique est quasipériodique. On obtient alors un analogue de la propriété 1.1.1 qui est vraie en toute dimensions est la suivante :

Propriété 1.1.2 (Durand [Dur99]). *Tout sous-shift non-vide contient une configuration quasipériodique.*

On peut même aller plus loin, avec la notion de minimalité :

Définition 1.1.17*Sous-shift minimal*

Un *sous-shift minimal* est un sous-shift X non vide tel qu'il n'existe pas de sous-shift $Y \subsetneq X$ non vide.

Intuitivement, un sous-shift minimal est un sous-shift dont tous les points contiennent les mêmes motifs. En effet, si un point contient un motif qui n'est pas dans certains autres, on peut interdire ce motif et obtenir un sous-shift non vide inclus dans celui-ci.

La propriété qui nous intéresse est la suivante :

Propriété 1.1.3 (Birkhoff [Bir12]). *Tout sous-shift non vide contient un sous-shift minimal.*

Le lien entre la quasipériodicité et la minimalité est dû au résultat ci-dessous.

Théorème 1.1.7. *Un sous-shift minimal ne contient que des configurations quasipériodiques.*

Preuve. Soit X un sous-shift minimal, supposons qu'il contienne une configuration c qui ne soit pas quasipériodique. Il existe alors un motif m qui n'apparaît pas dans des zones arbitrairement grandes de c . On a donc une suite croissante de motifs extensibles m_i dont aucun ne contient m , par compacité, on peut extraire de cette suite une suite convergeant vers une configuration c' , ne contenant pas m . Si l'on ajoute aux motifs interdits de X le motif M , celui-ci n'est donc pas vide, il contient donc strictement un autre sous-shift non vide. \square

Inversement, une configuration quasipériodique engendre un sous-shift minimal.

1.1.8 Conjugaison et invariants de conjugaison

Une des premières choses que l'on peut être amené à remarquer sur les sous-shifts est que le même sous-shift peut être défini par de nombreuses familles

de motifs interdits différentes. Par exemple, le premier exemple de sous-shift (figure 1.2) peut être défini par les deux ensembles de motifs interdits suivants sur l'alphabet $\{\blacksquare, \square\}$:

$$\mathcal{F}_1 = \left\{ \blacksquare\blacksquare, \square\square, \square, \blacksquare, \blacksquare\blacksquare, \blacksquare\square, \square\blacksquare, \blacksquare\square\blacksquare \right\}$$

$$\mathcal{F}_2 = \left\{ \blacksquare\blacksquare, \square\square, \blacksquare, \square \right\}$$

Plus généralement, on souhaite pouvoir déterminer si deux sous-shifts sont égaux ou s'ils ont les mêmes configurations à un recodage près. La notion d'isomorphisme qui capture parfaitement l'équivalence entre deux sous-shifts est la conjugaison, voir la définition 1.1.6.

La conjugaison correspond à un codage local réversible. Informellement, deux sous-shifts sont conjugués si l'on peut faire correspondre les configurations de l'un et de l'autre deux à deux, simplement par un renommage des couleurs dépendant uniquement d'un voisinage fini. Par exemple, les deux sous-shifts de la figure 1.7 sont conjugués.

La conjugaison peut être vue comme une relation d'équivalence et sépare les sous-shifts en classes où tous les éléments sont conjugués. Savoir si deux sous-shifts appartiennent à une même classe est le problème de la *classification des sous-shifts*.

Cette classification est intéressante parce que les éléments d'une même classe partagent de nombreuses propriétés et caractéristiques, et en particulier ont une même dynamique. Une propriété ou un objet associé à un sous-shift qui est préservé par conjugaison est appelé invariant par conjugaison :

Définition 1.1.18

Invariant de conjugaison

Un *invariant de conjugaison* est un objet $o(X)$ associé à un sous-shift X , tel que si X et Y sont deux sous-shifts conjugués, alors $o(X) = o(Y)$.

L'entropie est l'invariant de conjugaison le plus étudié :

Définition 1.1.19

Entropie

L'*entropie* $h(X)$ d'un sous-shift X est définie par

$$h(X) = \lim_{n \rightarrow \infty} \frac{\log \#\mathcal{E}_n(X)}{\#\mathcal{B}_n}$$

L'entropie est une mesure de la croissance du nombre de motifs de taille n qui apparaissent dans les points du sous-shift.

Les invariants de conjugaison permettent souvent de prouver de manière simple que deux sous-shifts ne sont pas conjugués. Ainsi, il suffit par exemple de prouver que deux sous-shifts n'ont pas la même entropie pour savoir qu'ils ne sont pas conjugués.

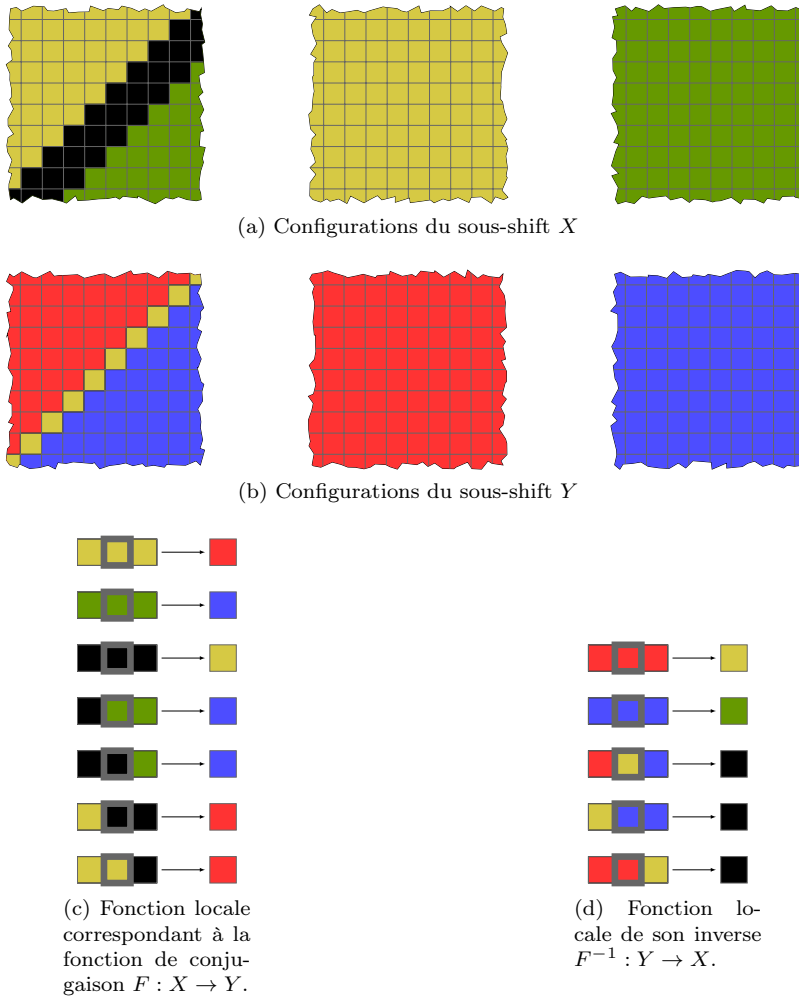


FIGURE 1.7 – Deux sous-shifts conjugués ainsi qu’une fonction de conjugaison et son inverse.

Théorème 1.1.8. *L’entropie est un invariant de conjugaison.*

Preuve. Soient X, Y deux sous-shifts conjugués. On peut supposer sans perte de généralité que la fonction de conjugaison et son inverse ont un rayon r , et on a donc :

$$\begin{aligned} \#\mathcal{E}_{n-r}(X) &\leq \#\mathcal{E}_n(Y) \leq \#\mathcal{E}_{n+r}(X) \\ \frac{\#\mathcal{E}_{n-r}(X)}{\#\mathcal{B}_{n-r}} \cdot \frac{\#\mathcal{B}_{n-r}}{\#\mathcal{B}_n} &\leq \frac{\#\mathcal{E}_n(Y)}{\#\mathcal{B}_n} \leq \frac{\#\mathcal{E}_{n+r}(X)}{\#\mathcal{B}_{n+r}} \cdot \frac{\#\mathcal{B}_{n+r}}{\#\mathcal{B}_n} \end{aligned}$$

Le passage à la limite, étant donné que $\lim_{n \rightarrow \infty} \frac{\#\mathcal{B}_{n \pm r}}{\#\mathcal{B}_n} = 1$, donne le résultat. \square

Avec l’entropie, on peut voir trivialement par exemple que le full-shift sur k symboles n’est pas conjugué au full-shift sur k' symboles quand $k \neq k'$.

En dimension 1 l'entropie d'un SFT est le logarithme du rayon spectral de la matrice le représentant. Les entropies y ont été caractérisées, par Lind [Lin84], comme étant les logarithmes de nombres de Perron multipliés par un rationnel positif. En dimensions supérieures, il existe également une caractérisation, qui n'est plus algébrique mais de nature calculatoire :

Théorème 1.1.9 (Hochman et Meyerovitch [HM10]). *Pour $d \geq 2$, la classe des entropies des SFTs (resp. sous-shifts sofiques) d -dimensionnels est la classe des réels positifs récursivement énumérables par la droite⁵.*

Il est aisé de voir que l'entropie est récursivement énumérable par la droite : il suffit de noter que $\#\mathcal{A}_n(X) \geq \#\mathcal{E}_n(X)$, puis de noter que remplacer le nombre de motifs extensibles par le nombre de motifs admissibles dans la formule définissant l'entropie ne change pas la limite. Comme le nombre de n -blocs admissibles est calculable, on a le résultat. La réalisation de tous les nombres réels récursivement énumérables par la droite comme entropies de SFTs est en revanche difficile.

Un second invariant intéressant et utile est le nombre de points périodiques :

Théorème 1.1.10. *La suite $(p_n(X))_{n \in \mathbb{N}^*}$ où $p_n(X)$ est le nombre de points de X de plus petite période n est un invariant de conjugaison.*

Preuve. Remarquons tout d'abord que l'invariant est bien défini, car il ne peut y avoir qu'un nombre fini de points périodiques de période n . Soient X et Y deux SFTs conjugués, et h leur fonction de conjugaison. Soit $x \in X$, si x est périodique de période p , alors $h(x)$ est aussi périodique de période p . Si de plus p est la plus petite période de x , alors c'est aussi la plus petite période de $h(x)$: en effet, si $h(x)$ avait une plus petite période que p , alors comme $x = h^{-1}(h(x))$, celle-ci serait aussi une période de x , ce qui est impossible. \square

En dimension 1, la suite $(p_n)_{n \in \mathbb{N}^*}$ correspond exactement à la suite $(\text{tr } A^n)_{n \in \mathbb{N}^*}$ des traces des puissances A , la matrice représentant le SFT. Nous donnerons une caractérisation de cette suite au chapitre 3 pour les dimensions supérieures à 2.

1.2 Calculabilité

Dans cette section nous commençons par introduire les notions de réduction pour les sous-ensembles de \mathbb{N} et la hiérarchie arithmétique. Ensuite nous donnons le premier lien entre SFTs et calculabilité : la preuve de l'indécidabilité de la pavabilité du plan. On pourra consulter le livre de Rogers [Rog87] pour une introduction en profondeur à la calculabilité.

Nous travaillerons toujours sur des machines de Turing à ruban semi-infini, où la tête peut aller dans trois directions : à gauche, à droite ou encore rester sur place.

On dira qu'une fonction ψ est *récursive* ou *calculable* s'il existe une machine de Turing la calculant. On note $\psi(x) \downarrow$ quand la fonction est définie en x (en

5. On rappelle qu'un nombre réel récursivement énumérable par la droite est un nombre h s'il existe une suite $(h_n)_{n \in \mathbb{N}}$ de rationnels tels que $h_n \rightarrow h$ et $h_n \geq h$. Cette notion correspond à la classe des nombres réels Π_1^0 de la hiérarchie arithmétique de nombres réels de Zheng et Weihrauch [ZW01].

particulier, on notera $M(x) \downarrow^t$ quand la machine M s'arrête au bout de t étapes) et $\psi(x) \uparrow$ lorsqu'elle ne l'est pas. Une fonction récursive est *totale* quand pour tout $n \in \mathbb{N}$, $\psi(n) \downarrow$.

Un *problème* désignera dans la suite un sous-ensemble de \mathbb{N} , une "solution" au problème étant un élément de ce sous-ensemble. Un problème est dit *récursif* si sa fonction caractéristique est récursive. On peut représenter un problème par un élément x de $\{0, 1\}^{\mathbb{N}}$: un élément x de $\{0, 1\}^{\mathbb{N}}$ est *récursif* s'il existe une machine de Turing qui prenant en entrée i retourne x_i . Un problème A est dit *récurivement énumérable* (resp. *co-récurivement énumérable*) s'il existe une machine de Turing qui énumère les éléments de A (resp. $\mathbb{N} \setminus A$).

Dans la suite, $(M_i)_{i \in \mathbb{N}}$ (resp. $(M_i^A)_{i \in \mathbb{N}}$) désignera une énumération acceptable des machines de Turing (resp. des machines de Turing à oracle avec oracle A).

1.2.1 Réductions et degrés Turing

Une fonction $\psi : \{0, 1\}^{\mathbb{N}} \rightarrow \{0, 1\}^{\mathbb{N}}$ est *récursive* s'il existe une machine de Turing à oracle M qui avec oracle x calcule $\psi(x)$, c'est à dire $M^x(i) = \psi(x)_i$. Il est intéressant de noter qu'une fonction $\psi : \{0, 1\}^{\mathbb{N}} \rightarrow \{0, 1\}^{\mathbb{N}}$ récursive est nécessairement continue (*use principle*).

Pour deux suites infinies $x, y \in \{0, 1\}^{\mathbb{N}}$ (que l'on peut également voir comme des fonctions caractéristiques de sous-ensembles A, B de \mathbb{N}), on dit que $x \leq_T y$ (x se réduit à y) si il existe une fonction récursive $\phi : \{0, 1\}^{\mathbb{N}} \rightarrow \{0, 1\}^{\mathbb{N}}$ tel que $\phi(y) = x$, il s'agit de la *réduction Turing*. Si $x \leq_T y$ et $x \geq_T y$, on note $x \equiv_T y$. La relation \equiv_T est une relation d'équivalence, et les classes d'équivalence par cette relation sont appelés *degrés Turing*, on note $\deg_T x$ le degré Turing de x . La relation \leq_T s'étend naturellement aux degrés Turing.

La réduction Turing permet d'ordonner les problèmes en fonction de leur difficulté. Cet ordre n'est pas total : il existe des degrés Turing incomparables. Le degré Turing le plus simple est $\mathbf{0}$, celui des ensembles récursifs.

Le *jump Turing* d'un ensemble A , noté A' est l'ensemble suivant

$$A' = \{n \mid M_n^A(n) \downarrow\}.$$

Pour n'importe quel problème A , on a $A <_T A'$. On peut définir inductivement $A^{(n+1)} = (A^{(n)})'$, avec $A^{(0)} = A$.

Il y a d'autres notions de réduction, et en particulier la réduction *many-one* \leq_m , qui est plus classique. Soient $A, B \subseteq \mathbb{N}$, on dit que $A \leq_m B$ s'il existe une fonction récursive totale $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $x \in A \Leftrightarrow f(x) \in B$ pour tout $x \in \mathbb{N}$. Si l'on voit A, B comme des éléments x, y de $\{0, 1\}^{\mathbb{N}}$, alors $x \leq_m y$ si il existe une fonction récursive totale f telle que $x_i = y_{f(i)}$. Il est clair que si $x \leq_m y$, alors $x \leq_T y$, le contraire n'étant pas vrai. On peut aussi définir des *degrés many-one* de manière analogue aux degrés Turing, nous n'en aurons cependant pas l'usage ici.

On dira que deux ensembles A, B sont *récurivement isomorphes* s'il existe une fonction $f : A \rightarrow B$ bijective récursive dont l'inverse est récursif.

1.2.2 Hiérarchie arithmétique

La hiérarchie arithmétique est une classification des problèmes par rapport à leur caractérisation logique :

Définition 1.2.1

Hiérarchie Arithmétique

Un ensemble $S \subseteq \mathbb{N}$ est Σ_n^0 , s'il existe un prédicat récursif ϕ tel que

$$x \in S \Leftrightarrow \exists x_1, \forall x_2, \dots Qx_n, \psi(x, x_1, \dots, x_n),$$

où Q est un \forall si n est impair et un \exists sinon. Un ensemble est Π_n^0 si c'est le complémentaire d'un ensemble Σ_n^0 et Δ_n^0 s'il est à la fois Σ_n^0 et Π_n^0 .

En particulier, un ensemble S est Π_1^0 s'il existe une machine de Turing M telle que

$$x \in S \Leftrightarrow M(x) \text{ ne s'arrête pas.}$$

Inversement, un ensemble S est Σ_1^0 s'il existe une machine de Turing M telle que

$$x \in S \Leftrightarrow M(x) \text{ s'arrête.}$$

Les ensembles Σ_1^0 et Π_1^0 sont donc respectivement les ensembles récursivement énumérables et co-récursivement énumérables.

On peut d'ailleurs définir la hiérarchie à partir de relativisations et donc uniquement en termes de machines de Turing : un ensemble S est Σ_n^0 (resp. Π_n^0) s'il existe un ensemble A Π_n^0 (resp. Σ_n^0) et une machine de Turing M telle que M avec oracle A s'arrête (resp. ne s'arrête pas) sur l'entrée x si et seulement si $x \in S$.

À l'intérieur d'une même classe Π_n^0 ou Σ_n^0 , il existe des problèmes plus durs que les autres : les problèmes complets. Les problèmes complets sont des problèmes qui permettent de résoudre tous ceux de la même classe.

Définition 1.2.2

Problèmes complets

Un problème P est dit Σ_n^0 -dur (resp. Π_n^0 -dur) pour la réduction many-one si pour tout problème A qui est Σ_n^0 , on a $A \leq_m P$. Il est dit Σ_n^0 -complet si en plus il est Σ_n^0 (resp. Π_n^0).

En particulier, on peut voir directement que l'arrêt, $\mathbf{0}'$, est complet pour la classe Σ_1^0 , et son complémentaire pour la classe Π_1^0 . Pour chaque niveau de la hiérarchie il y a des problèmes complets. Le problème complet canonique pour Σ_n^0 est $\mathbf{0}^{(n)}$:

Théorème 1.2.1. $\mathbf{0}^{(n)}$ est Σ_n^0 -complet.

Un exemple simple de problème complet est TOTAL : TOTAL est l'ensemble des machines de Turing qui s'arrêtent sur toutes les entrées.

Théorème 1.2.2. *TOTAL est Π_2^0 -complet.*

Preuve. Pour voir que total est dans Π_2^0 , il suffit de remarquer que l'on peut exprimer pour une machine M le fait qu'elle est totale par la formule suivante :

$$\forall x \exists t M(x) \downarrow^t$$

On peut bien entendu décider $M(x) \downarrow^t$: il suffit de simuler t étapes de calcul de M . On va maintenant prouver que le problème est Σ_2^0 -dur : Soit $S = \{n \mid \forall x, \exists y, \psi(n, x, y)\}$ un ensemble Σ_2^0 -dur, à chaque $n \in \mathbb{N}$, on associe une machine \mathcal{M}_n qui sur l'entrée x teste en parallèle pour tous les y si $\psi(n, x, y)$ et s'arrête lorsqu'elle en a trouvé un. Ainsi, $\mathcal{M}_n(x) \downarrow$ si et seulement si $\exists y, \psi(n, x, y)$, et \mathcal{M}_n est totale si et seulement si $\forall x, \exists y, \psi(n, x, y)$. \square

Dans la suite, nous allons nous servir du problème COFIN dans certaines réductions : COFIN est l'ensemble des machines de Turing qui ne s'arrêtent pas sur un nombre fini d'entrées.

Théorème 1.2.3. *COFIN est Σ_3^0 -complet.*

La preuve de ce résultat peut être trouvée dans le livre de Kozen [Koz06].

Tous les problèmes naturels ne sont pas dans la hiérarchie arithmétique : étant donné un SFT et un symbole, savoir si ce SFT contient une configuration contenant une infinité de fois ce symbole est un problème dit analytique, voir Harel [Har84].

1.2.3 Indécidabilité de la pavabilité du plan

On a évoqué plus tôt, en section 1.1.2, le fait que savoir si un SFT est vide est indécidable, on donne ici une preuve ultérieure à celle de Berger, due à Robinson [Rob71]. Celle-ci est un peu plus simple et surtout sera à la base de certaines constructions dans la suite.

Théorème 1.2.4 (Berger [Ber64]). *L'ensemble $\{\mathcal{F} \mid \mathcal{X}_{\mathcal{F}} = \emptyset\}$ est Σ_1^0 -complet.*

Il est clair que le problème est dans Σ_1^0 : étant donné \mathcal{F} , si $\mathcal{X}_{\mathcal{F}}$ est vide, alors il existe k tel qu'il n'existe aucun motif admissible de support \mathcal{B}_k . La preuve de la dureté est plus ardue, nous la donnons dans la suite (théorème 1.2.5), mais nous expliquons d'abord comment coder des calculs de machine de Turing dans les SFTs et le jeu de tuiles de Robinson qui constituent le coeur de la preuve.

Liens entre SFTs et machines de Turing

Nous allons maintenant expliquer un codage qui explique pourquoi les SFTs de dimension supérieure à 2 et les machines de Turing sont particulièrement liés.

Le ruban d'une machine de Turing, à une étape de temps t , peut être vu comme un mot infini. En rajoutant l'état et la position de la tête, on a donc toute l'information sur la machine à cette étape de temps et on peut en déduire l'état au temps $t + 1$. On peut donc représenter le ruban et l'état de la machine au temps t par le mot suivant :

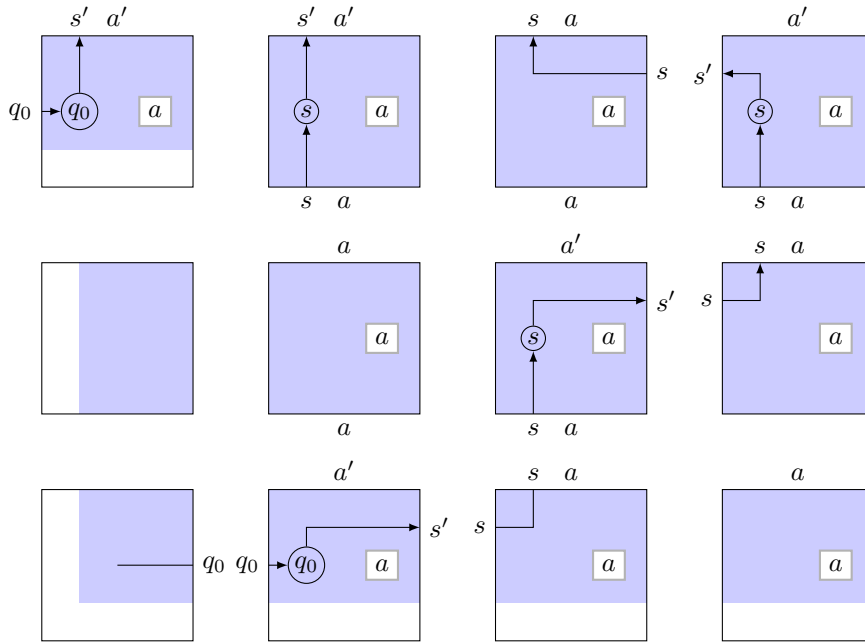
$$a_0^t a_1^t \dots a_i^t (q, a_{i+1}^t) a_{i+2}^t \dots$$

où q est l'état dans lequel est la machine à cette étape et la tête est sur la case où se trouve q . Entre t et $t + 1$, la tête bouge au plus de une case et la lettre dans une case ne dépend que de la même case au temps précédent et de ses deux voisines.

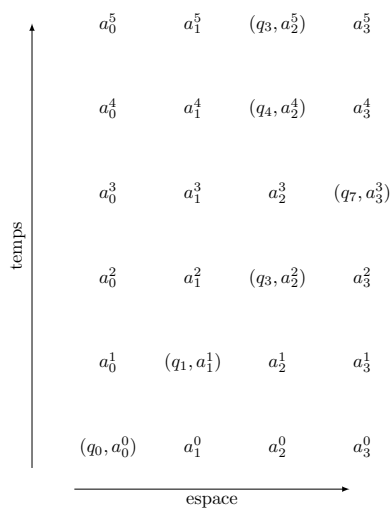
Le diagramme espace temps d'une machine de Turing peut donc être vérifié à l'aide de contraintes locales. Ainsi à une machine de Turing donnée, on peut associer un jeu de tuiles. Ce jeu de tuiles est calculable en temps linéaire à partir de la description de la machine de Turing. L'idée est que chaque tuile contient un morceau du ruban. On peut voir en figure 1.8 le jeu de tuiles associé à une machine de Turing et la manière dont un diagramme espace temps est vérifié par un jeu de tuiles.

Ainsi si l'on se donne une machine de Turing et le jeu de tuiles associé, ainsi qu'une tuile blanche, et que l'on force la tuile en $(0, 0)$ à être la tuile qui initie le calcul, celle dont les bords gauche et bas sont blancs, on ne peut paver le plan que si la machine de Turing ne s'arrête pas.

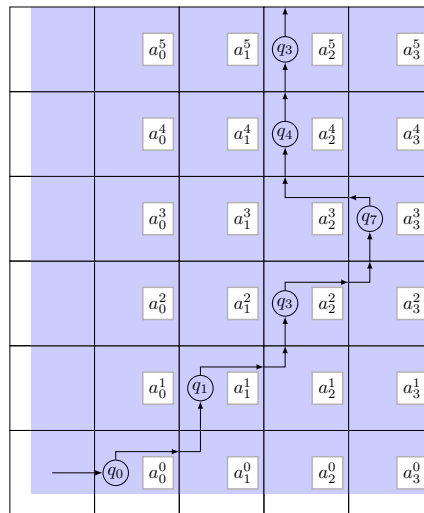
On ne peut néanmoins pas déduire directement l'indécidabilité de la pavabilité du plan de ce codage : en effet, on a besoin de forcer l'apparition de la tuile initiant le calcul pour que celui-ci ait lieu. Ainsi, si on prenait le jeu de tuiles tel quel, on obtiendrait des pavages ne correspondant à aucun calcul : par exemple un pavage avec uniquement des tuiles contenant le ruban pourrait paver le plan à partir de n'importe quel mot biinfini sur l'alphabet de la machine. On doit donc trouver un moyen de forcer les seuls pavages valides à être ceux où un calcul valide apparaît. Ceci va être fait grâce à un jeu de tuiles apériodique.



(a) Tuiles permettant d'encoder une machine de Turing. Les tuiles sont données par la fonction de transition de la machine $\delta(s, a) = (s', a', d)$ où d détermine de quel côté la flèche sort de la tuile.



(b) Diagramme espace temps d'une machine de Turing.



(c) Pavage valide par le jeu de tuiles de la machine de Turing.

FIGURE 1.8 – Comment coder des machines de Turing dans des pavages. Seules les évolutions valides de la machine de Turing peuvent le plan sans erreur.

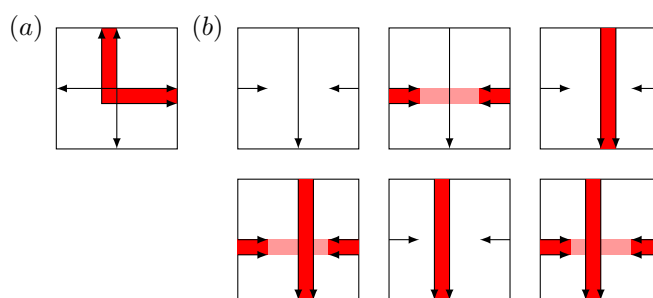


FIGURE 1.9 – Le jeu de tuiles de Robinson : en (a) les croix et en (b) les bras. Selon dans quel sens est la flèche principale (la seule qui soit continue d'un bord à un autre), un bras est soit vertical soit horizontal.

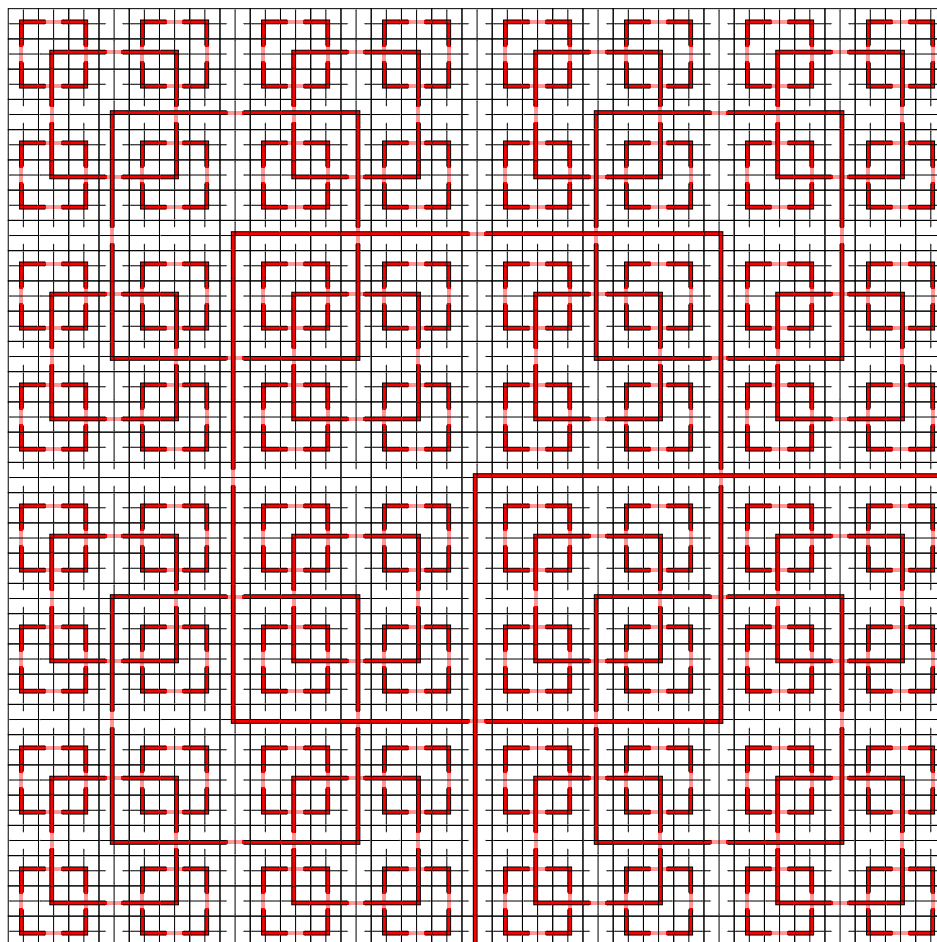


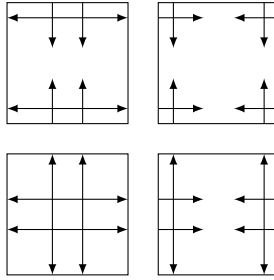
FIGURE 1.10 – Une partie d'un pavage obtenu à l'aide du jeu de tuiles de Robinson. On peut remarquer la structure hiérarchique formée de carrés imbriqués.

Jeu de tuiles de Robinson

Le jeu de tuiles de Robinson est apériodique et va permettre de faire apparaître du calcul dans tous les pavages. Ce jeu de tuiles a été introduit par Robinson [Rob71], puis Allauzen et Durand [AD01] en ont ensuite simplifié la preuve.

On va représenter le jeu de tuiles de Robinson à l'aide de tuiles où les couleurs ont été représentées par des flèches, deux tuiles pouvant être voisines si les flèches sur leur bord commun se prolongent et vont dans le même sens. On ne donne ici que les tuiles et l'idée de la preuve que la construction forme bien un jeu de tuiles apériodique non-vide.

Le jeu de tuiles est constitué de deux couches, les tuiles de base, en figure 1.9, et une deuxième couche, les tuiles de parité :



Dans un pavage à l'aide des tuiles de parité, les seuls pavages possibles sont ceux où les tuiles alternent horizontalement et verticalement dans le même ordre que ci-dessus.

On va imposer certaines règles de superposition entre les deux couches. La tuile de parité d'en bas à gauche ne peut être superposée qu'à une croix (définie en figure 1.9), tandis que celle d'en bas à droite (resp. haut à gauche) ne peut être superposée qu'à un bras vertical (resp. horizontal).

Les pavages obtenus ont une structure hiérarchique : des carrés de côté $2^k + 1$ ayant une croix au centre qui est le coin d'un carré de côté $2^{k+1} + 1$, et ainsi de suite. La structure hiérarchique des pavages garantit leur apériodicité. On peut voir en figure 1.10 un extrait de n'importe quel pavage par ce jeu de tuiles.

Il reste maintenant à incorporer le calcul dans ce jeu de tuiles. L'idée est que chaque carré va contenir un calcul partiel de la machine de Turing. On souhaite néanmoins éviter que plusieurs calculs ne se superposent. La première astuce est de remarquer que les carrés de côté $2^{2k} + 1$ (resp. $2^{2k+1} + 1$) n'intersectent pas les carrés de côté $2^{2k'}$ (resp. $2^{2k'+1} + 1$), c'est à dire que les carrés d'ordre pair (resp. impair) dans la hiérarchie n'intersectent pas d'autre carré d'ordre pair (resp. impair). Afin de pouvoir identifier la parité des niveaux, on introduit une nouvelle couleur en plus au lieu d'une seule pour les carrés, les nouvelles tuiles sont en figure 1.11. Ceux-ci peuvent donc maintenant être soit rouges soit bleus, un carré rouge n'intersecte pas de carré rouge et un carré bleu n'intersecte pas de carré bleu. On pourra se référer à la figure 1.12.

La dernière étape de la construction permet de montrer la dernière partie du théorème 1.2.4 :

Théorème 1.2.5. *Le problème $\{\mathcal{F} \mid \mathcal{X}_{\mathcal{F}} = \emptyset\}$ est Σ_1^0 -dur.*

Preuve. La réduction va se faire en prenant le problème $\mathbf{0}' = \{M \mid M \downarrow\}$. On veut donc associer à une machine de Turing M un jeu de tuiles τ_M tel que $\mathcal{X}_{\tau_M} = \emptyset$ si

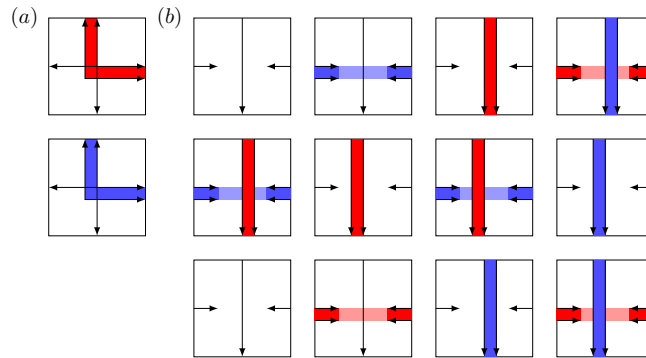


FIGURE 1.11 – Les tuiles de base du jeu de tuiles de Robinson avec deux couleurs.

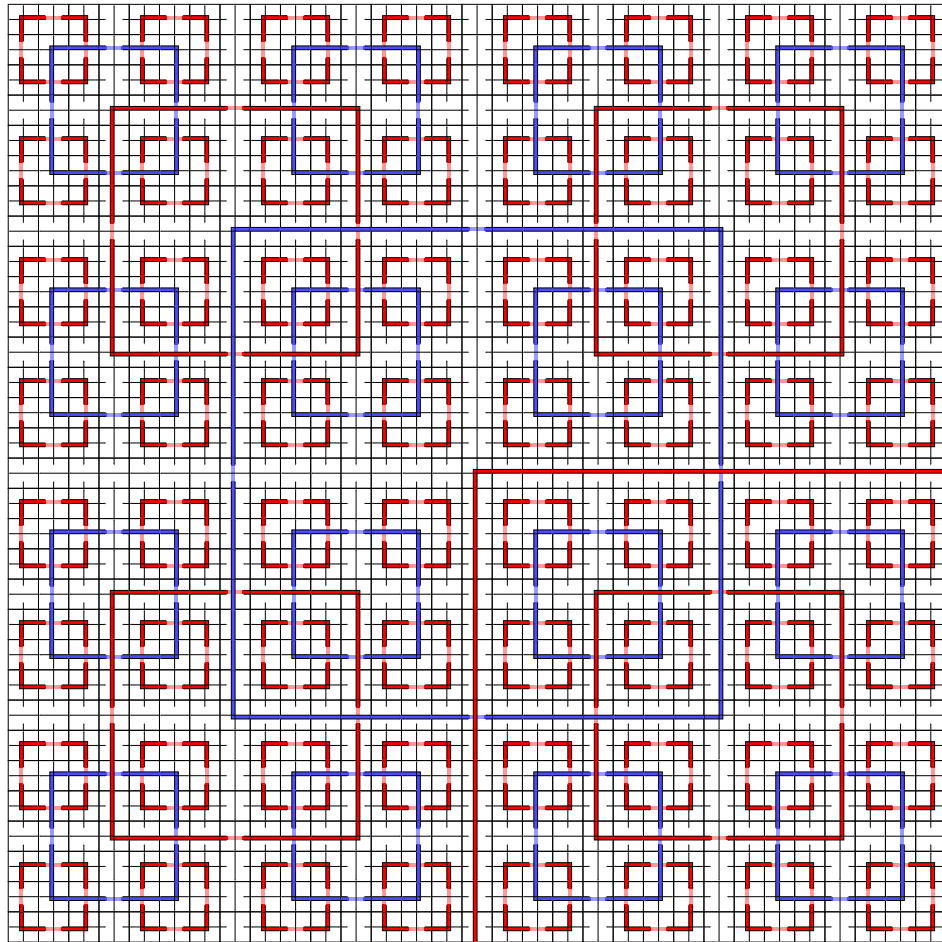


FIGURE 1.12 – Une partie d'un pavage à l'aide du jeu de tuiles de Robinson avec les deux couleurs, les carrés de même couleur ne s'intersectent jamais. Les carrés bleus sont de taille $2^{2k+2} + 1$ et les carrés rouges de taille $2^{2k+1} + 1$, avec $k \in \mathbb{N}$.

et seulement si $M \downarrow$.

En utilisant le jeu de tuiles de Robinson défini précédemment, il suffit de faire en sorte que chaque carré bleu contienne un calcul partiel de la machine de Turing M , et d'interdire l'état d'arrêt d'apparaître.

La difficulté est d'éviter une superposition infinie de calculs, étant donné qu'à l'intérieur d'un carré bleu il y a de nombreux carrés bleus plus petits. La régularité des pavages permet néanmoins de ne mettre du calcul que dans les zones du grand carré qui ne sont alignées ni verticalement ni horizontalement avec les plus petits carrés. Ces zones doublent de taille à chaque niveau, l'espace/temps disponible croît donc avec le niveau auquel on se trouve. Si la machine s'arrête, il y aura donc une erreur de pavage à un des niveaux. \square

Chapitre 2

Classes Π_1^0 et sous-shifts

Sommaire

2.1	Classes Π_1^0	34
2.1.1	Degrés Medvedev et Muchnik	35
2.1.2	Des classes Π_1^0 aux pavages	37
2.1.3	Degrés Medvedev des SFTs	37
2.2	Degrés Turing	38
2.2.1	Une grille irrégulière	38
2.2.2	Degrés Turing des classes Π_1^0 et des SFTs	44
	Sous-shifts unidimensionnels	45
	Sous-shifts de dimensions supérieures	48

Dans ce chapitre, nous tentons de mieux comprendre la structure calculatoire des SFTs. La première étape est de comprendre à quel point les configurations d'un SFT peuvent être compliquées, au sens calculatoire du terme : plus le degré Turing d'une configuration est élevé, plus elle est compliquée. Il est clair qu'il y a des SFTs avec des configurations ayant un degré Turing arbitrairement élevé : le full-shift contient toutes les configurations, il contient donc des configurations arbitrairement compliquées mais également des configurations très simples. Une première question est donc plutôt de savoir s'il existe des SFTs dont tous les points sont de degré Turing élevé. Une réponse à cette question a été donnée par Hanf [Han74], Myers [Mye74], qui ont prouvé le résultat suivant :

Théorème 2.0.6 (Hanf [Han74], Myers [Mye74]). *Il existe un SFT X ne contenant aucun point récursif.*

Mais on sait également que les SFTs contiennent toujours un point de degré Turing strictement inférieur à $\mathbf{0}'$, voir Jockusch et Soare [JS72].

Les sous-shifts de type fini, effectifs et sofiqes sont des classes Π_1^0 , des objets très étudiés de la calculabilité, voir Cenzer et Remmel [CR98], Simpson [Sim11a] par exemple. Les classes Π_1^0 sont les ensembles de $\{0, 1\}^{\mathbb{N}}$ effectivement clos, c'est à dire les ensembles d'oracles sur lesquels une machine de Turing donnée ne s'arrête pas. La compréhension de la structure calculatoire des SFTs passe donc par la comparaison avec les classes Π_1^0 en général. Le premier résultat dans ce sens est dû à Simpson [Sim11b], qui a caractérisé les degrés de difficulté

des SFTs à l'aide des degrés Medvedev et Muchnik, montrant que c'étaient exactement les mêmes que ceux des classes Π_1^0 . Miller [Mil12] est arrivé à la même caractérisation en termes de degrés Medvedev/Muchnik pour les sous-shifts effectifs en dimension 1.

Cependant, les degrés Muchnik et Medvedev ne donnent pas d'information sur toute la structure de la classe Π_1^0 , ils ne donnent de l'information que sur les points les plus faciles. Donc pour avoir une compréhension plus complète de la structure des SFTs, il faut également regarder la structure de leurs degrés Turing. Nous tentons d'apporter dans ce chapitre des réponses aux questions suivantes :

- Quels sont les ensembles de degrés Turing des SFTs ?
- Est-ce que ce sont exactement les mêmes que ceux des classes Π_1^0 de $\{0, 1\}^{\mathbb{N}}$?
- Existe-t'il pour chaque classe Π_1^0 de $\{0, 1\}^{\mathbb{N}}$ un SFT qui lui est récursivement isomorphe ?

En dimension 1, des réponses partielles à certaines de ces questions ont été données par Cenzer, Dashti et King [CDK08], Cenzer, Dashti, Toska et Wyman [Cen+12, Cen+10] dans le cas des sous-shifts dénombrables et effectifs en les catégorisant par rang de Cantor-Bendixson, voir Kuratowski [Kur66]. En effet, le cas des SFTs et des sous-shifts sofiques n'est pas intéressant : soit le sous-shift est vide, soit tous ses points sont récursifs, soit il est isomorphe à $\{0, 1\}^{\mathbb{N}}$.

Une autre point de vue a été adopté par Durand, Levin et Shen [DLS08], qui se sont intéressés à la complexité algorithmique des motifs apparaissant dans les points d'un SFTs de dimension $d \geq 2$, prouvant que tout SFT a une configuration dont tous les motifs sont simples au sens de Kolmogorov.

Nous tentons d'apporter des réponses complètes à ces trois questions dans ce chapitre. Dans un premier temps, en section 2.1, nous rappelons la définition de classe Π_1^0 et de degrés Medvedev et Muchnik. Nous y donnons également le lien entre classes Π_1^0 et SFTs, ainsi que la preuve de Simpson. Ensuite, en section 2.2, nous étudions les ensembles de degrés Turing des SFTs. Nous commençons par prouver (section 2.2.1) que pour toute classe Π_1^0 de $\{0, 1\}^{\mathbb{N}}$, il existe un SFT qui lui est quasi-isomorphe. Pour prouver ce résultat, nous introduisons une nouvelle construction, basée sur une grille irrégulière, qui nous permet d'insérer du calcul dans un SFT dont toutes les configurations sont récursives, de manière à ce que le SFT de base n'ajoute pas de complexité au calcul. Enfin en section 2.2.2 nous nous intéressons à la structure des degrés Turing des sous-shifts et prouvons que les ensembles de degrés Turing des SFTs ne contenant aucun point récursif contiennent toujours un cône de degrés Turing : un degré ainsi que tous les degrés qui lui sont plus grands.

2.1 Classes Π_1^0

Si un jeu de tuiles, une collection de motifs interdits, est une donnée finie, il n'en est pas de même des points du SFT qui lui est associé. À un recodage récursif près, on peut voir les configurations des SFTs comme des points de $\{0, 1\}^{\mathbb{N}}$. On peut alors vérifier si une configuration c appartient à un SFT X à l'aide d'une machine de Turing M de la manière suivante : la machine va vérifier à chaque coordonnée, selon un bon ordre, s'il y a un motif interdit. Si tel est le cas, la machine s'arrête, dans le cas contraire elle continue à la coordonnée

suivante. On a alors :

$$c \in X \quad \Leftrightarrow \quad M(c) \uparrow$$

Ce mode de reconnaissance peut être mis en parallèle avec les ensembles Π_1^0 que l'on a vu précédemment à la section 1.2.2 : on accepte uniquement les éléments sur lesquels la machine ne s'arrête pas. La principale différence par rapport aux ensembles Π_1^0 est que les entrées sont maintenant des éléments de $\{0, 1\}^{\mathbb{N}}$ et non plus des entiers. On appelle les sous-ensembles de $\{0, 1\}^{\mathbb{N}}$ reconnus de cette manière des *classes* Π_1^0 .

Définition 2.1.1

Classe Π_1^0

Un ensemble $S \subseteq \{0, 1\}^{\mathbb{N}}$ est une *classe* Π_1^0 si et seulement s'il existe une machine de Turing à oracle M telle que :

$$x \in S \quad \Leftrightarrow \quad M \text{ ne s'arrête pas sur l'oracle } x \\ \text{en partant du ruban vide.}$$

Les classes Π_1^0 sont également appelées *ensembles effectivement clos*.

Comme exemples de classes Π_1^0 , on a par exemple :

- L'ensemble des suites telles que si i est une puissance de 2, alors $x_i = 1$: s'il y a une erreur, on peut la détecter en temps fini, mais s'il n'y en a pas, la machine de Turing ne s'arrêtera jamais.
- L'ensemble des complétions d'une théorie T récursivement énumérable : un ensemble de formules qui rendent la théorie T complète. En effet étant donné un candidat à la completion, il faut vérifier que T avec sa complétion est consistante, mais également pour toute formule ϕ que celle-ci est soit vraie, soit fausse dans la théorie complétée.

Les classes Π_1^0 sont également appelées *problèmes de masse* (*mass problems* en anglais), voir Simpson [Sim11a] et ont été intensivement étudiées, on pourra également se référer au futur livre de Cenzer et Remmel [CR] ou à leur article de synthèse [CR98].

Théorème 2.1.1. *Un sous-shift sofique/effectif/de type fini est une classe Π_1^0 .*

Preuve. Comme la classe des sous-shifts effectifs englobe les deux autres classes, il suffit de donner la preuve dans ce cas. Soit X un sous-shift effectif, et M la machine qui énumère ses motifs interdits. La machine reconnaissant les points du sous-shift lance M et vérifie en parallèle si aucun des motifs générés par M n'apparaît dans l'oracle c : la machine garde une liste des motifs déjà générés et revérifie à chaque motif interdit supplémentaire toutes les coordonnées déjà vérifiées et une coordonnée supplémentaire. \square

2.1.1 Degrés Medvedev et Muchnik

On a vu précédemment les degrés Turing comme évaluation de la difficulté d'un problème. Ceux-ci ne permettent cependant pas de caractériser directement la difficulté des problèmes de masse. En effet, pour les problèmes de masse, il faudrait considérer l'ensemble formé par les degrés Turing de chacun des

membres. L'ensemble des degrés Turing des membres est une mesure très fine de la complexité du problème de masse, ce n'est cependant pas un objet facile à manipuler. Afin d'étudier la complexité des problèmes de masse, ont donc été introduits les degrés Medvedev et Muchnik, [Med55, Muc63].

Définition 2.1.2*Degré Medvedev*

Soit $S_1, S_2 \subseteq \{0, 1\}^{\mathbb{N}}$, alors $S_1 \leq_s S_2$ s'il existe une fonction récursive $f : S_2 \rightarrow S_1$. On note \equiv_s la relation d'équivalence associée. Les *degrés Medvedev* sont alors les classes d'équivalence de \equiv_s , on note $\text{deg}_s S$ le degré Medvedev d'une classe S .

Un problème de masse S_1 se réduit Medvedev à un problème S_2 s'il existe une procédure pour exhiber une solution du problème S_1 à partir d'une solution du problème S_2 , c'est à dire si à partir de tout point de S_2 on peut calculer un point de S_1 . Le degré Muchnik est la version non-uniforme du degré Medvedev :

Définition 2.1.3*Degré Muchnik*

Soit $S_1, S_2 \subseteq \{0, 1\}^{\mathbb{N}}$, on note $S_1 \leq_w S_2$ si

$$\forall x \in S_2, \exists y \in S_1, y \leq_T x.$$

On note \equiv_w la relation d'équivalence associée. Les *degrés Muchnik* sont alors les classes d'équivalence de \equiv_w , on note $\text{deg}_w S$ le degré Muchnik d'une classe S .

Un excellent survey sur les degrés Medvedev et Muchnik a été écrit par Hinman [Hin12].

Il est clair que pour deux problèmes de masse S_1, S_2 :

- Si $S_1 \leq_s S_2$, alors $S_1 \leq_w S_2$: ceci explique la notation $_s$ et $_w$, l'une étant forte, *strong*, l'autre étant faible, *weak*.
- Si l'ensemble des degrés Turing de S_1 est inclus dans celui de S_2 , alors $S_2 \leq_w S_1$.

Les degrés Medvedev et Muchnik forment des treillis distributifs bornés, dont les bornes sup et inf sont les suivantes, pour $S_1, S_2 \subseteq \{0, 1\}^{\mathbb{N}}$:

$$S_1 \wedge S_2 = \{0f \mid f \in S_1\} \cup \{1g \mid g \in S_2\}$$

$$S_1 \vee S_2 = \{h \mid f \in S_1, g \in S_2, h(2i) = f(i), h(2i+1) = g(i)\}$$

dont les bornes inférieures et supérieures sont $\mathbf{0} = \text{deg}(\{0000\dots\})$ et $\infty = \text{deg}(\emptyset)$. Dans le cas des classes Π_1^0 non-vides, ce sont toujours des treillis distributifs bornés. La borne inf reste la même, mais la borne sup devient $\text{deg}(DNR_2)$, avec $DNR_2 = \left(x \in \{0, 1\}^{\mathbb{N}} \mid x_i \neq M_i(i)\right)$, $(M_i)_{i \in \mathbb{N}}$ étant une énumération acceptable des machines de Turing.

2.1.2 Des classes Π_1^0 aux pavages

On a vu précédemment que chaque SFT était une classe Π_1^0 , mais les classes Π_1^0 sont-elles isomorphes aux SFTs ? Un premier résultat, facile, est le suivant :

Lemme 2.1.1. *Pour toute classe Π_1^0 S , il existe un SFT X et un symbole t , tel que l'ensemble des configurations de X ayant t à l'origine est récursivement isomorphe à S .*

Preuve. Soit M une machine de Turing reconnaissant S , on peut l'encoder dans un jeu de tuiles, comme on l'a vu précédemment (voir figure 1.8a). On force la tuile en $(0, 0)$ à être celle commençant le calcul. On rajoute une information sur chaque tuile contenant le ruban : un oracle. C'est à dire qu'on rajoute simplement sur chacune de ces tuiles un symbole 0 ou 1, tout en forçant celui-ci à rester le même sur chaque colonne. Ainsi, de manière non déterministe, tous les oracles peuvent apparaître, mais seuls ceux sur lesquels la machine de Turing ne s'arrête pas ne provoqueront pas d'erreur. L'isomorphisme est alors le suivant : à un membre s de S , on fait correspondre le point x , tel que $x_{(i,0)}$ soit une tuile de bord bas, contenant s_i comme oracle, les symboles de x en n'importe quelle coordonnée sont alors déterminés par le calcul. Dans l'autre sens, il suffit de récupérer le contenu de l'oracle à la ligne d'ordonnée 0. \square

Ce résultat n'est pas très satisfaisant dans la mesure où l'on s'est affranchi de la caractéristique principale des SFTs : l'invariance par translation. Dans la suite nous allons prouver des résultats liant les degrés de difficulté des classes Π_1^0 et des SFTs tout en respectant les particularités des SFTs.

2.1.3 Degrés Medvedev des SFTs

Le lien entre les degrés Medvedev des SFTs et des classes Π_1^0 a été étudié pour la première fois dans Simpson [Sim11b], avec comme résultat principal le théorème suivant :

Théorème 2.1.2 (Simpson [Sim11b]). *Les degrés Medvedev des classes Π_1^0 sont les mêmes que ceux des SFTs de dimension 2 (et donc aussi > 2). Plus précisément, pour toute classe Π_1^0 , il existe un SFT de même degré, et inversement.*

La preuve de ce théorème est une modification mineure de la preuve de Hanf [Han74] et Myers [Mye74].

Idée de preuve. Le premier sens est facile, tout SFT étant une classe Π_1^0 . Dans l'autre sens, on part d'une classe Π_1^0 S , et on exhibe un SFT dont on peut extraire de chaque point un membre de S et pour lequel à partir de chaque membre de S on peut produire un point.

Le jeu de tuiles de Robinson permet d'avoir des calculs arbitrairement longs d'une machine de Turing. On souhaiterait utiliser la méthode du lemme 2.1.1, mais le problème est que sans modification du jeu de tuiles, chaque diagramme espace-temps partiel apparaissant dans un point peut avoir un oracle (partiel) différent. Myers [Mye74] explique cependant comment on peut synchroniser le ruban de départ de la machine de Turing pour tous les carrés à l'intérieur des pavages par le jeu de tuiles de Robinson. On peut alors récupérer l'oracle à partir d'un point en regardant des carrés de plus en plus grands, ce qui prouve un sens. Inversement, on peut aussi aisément construire un point à partir d'un membre de la classe Π_1^0 et d'un des points calculables que contient le SFT de Myers. \square

Il n'y a cette fois-ci plus d'isomorphisme : le SFT de Myers sans le calcul contient 2^{\aleph_0} points, et lorsque l'on ajoute le calcul, chaque membre de la classe Π_1^0 apparaîtra sur chacun de ces 2^{\aleph_0} points.

Pour la dimension 1, on arrive à la même caractérisation, mais en prenant les sous-shifts effectifs :

Théorème 2.1.3 (Miller [Mil12]). *Les degrés Medvedev des classes Π_1^0 sont les mêmes que ceux des sous-shifts effectifs de dimension 1. Plus précisément, pour toute classe Π_1^0 , il existe un sous-shift effectif de même degré, et inversement.*

2.2 Degrés Turing

Le problème de la construction précédente est qu'à un membre de la classe Π_1^0 S correspondent une infinité non dénombrable de points du SFT X . La fonction $f : X \rightarrow S$ amène donc des points *très complexes* sur des points *beaucoup plus simples*. L'idéal serait de réussir à avoir une construction qui conserve le degré de difficulté de chaque membre, et donc d'avoir plutôt un isomorphisme récursif. Comme il y a des classes Π_1^0 dénombrables, si l'on souhaite une construction générique il nous faut un SFT dénombrable dans lequel on peut introduire des calculs de machines de Turing. Cependant, ce SFT aura nécessairement des configurations dans lesquelles on ne peut pas mettre de calcul : étant dénombrable, il a nécessairement des configurations périodiques (on pourra consulter Durand [Dur99]), qui ne peuvent donc pas contenir de calcul.

Nous allons donc prouver le théorème suivant :

Théorème 2.2.1

Quasi-isomorphisme

Pour toute classe Π_1^0 $S \subseteq \{0, 1\}^{\mathbb{N}}$, il existe un SFT X , tel que S est récursivement isomorphe à $X \setminus C$ quotienté par \mathbb{Z}^2 , où C est un ensemble récursif de configurations récursives.

Ce résultat est optimal : on verra en effet dans la suite qu'il est impossible d'avoir un isomorphisme. Une des conséquences de ce théorème concerne les ensembles de degrés Turing : il assure que la classe Π_1^0 S et $X \setminus C$ ont les mêmes ensembles de degrés Turing et donc que S et X ont les mêmes ensembles de degrés Turing, à $\mathbf{0}$ près.

2.2.1 Une grille irrégulière

Afin d'arriver à ce résultat, nous avons besoin d'une grille infinie pour y mettre du calcul. Il faut aussi que toutes les configurations où il n'y a pas de calcul soient très simples, c'est à dire récursives. Il faut en particulier s'arranger pour que quand cette grille apparaît dans un point, elle contienne forcément un calcul correct.

Pour cela, on doit faire en sorte que dans tout point où la grille infinie apparaît il y ait un "coin", une tuile spéciale permettant de commencer le calcul. Une grille simple comme en figure 2.1a amènerait par compacité des configurations comme en figure 2.1b et ne permettrait donc pas d'atteindre notre but.

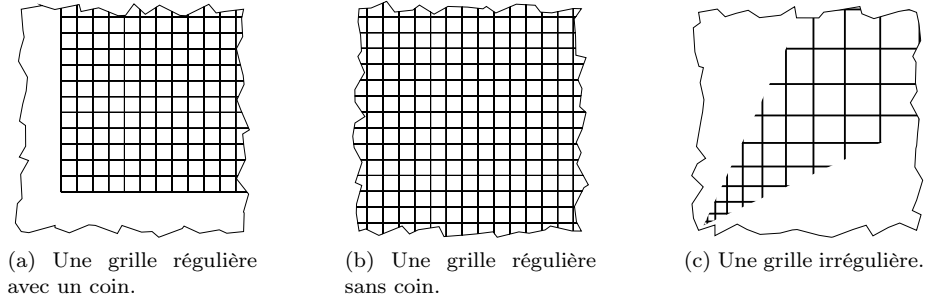


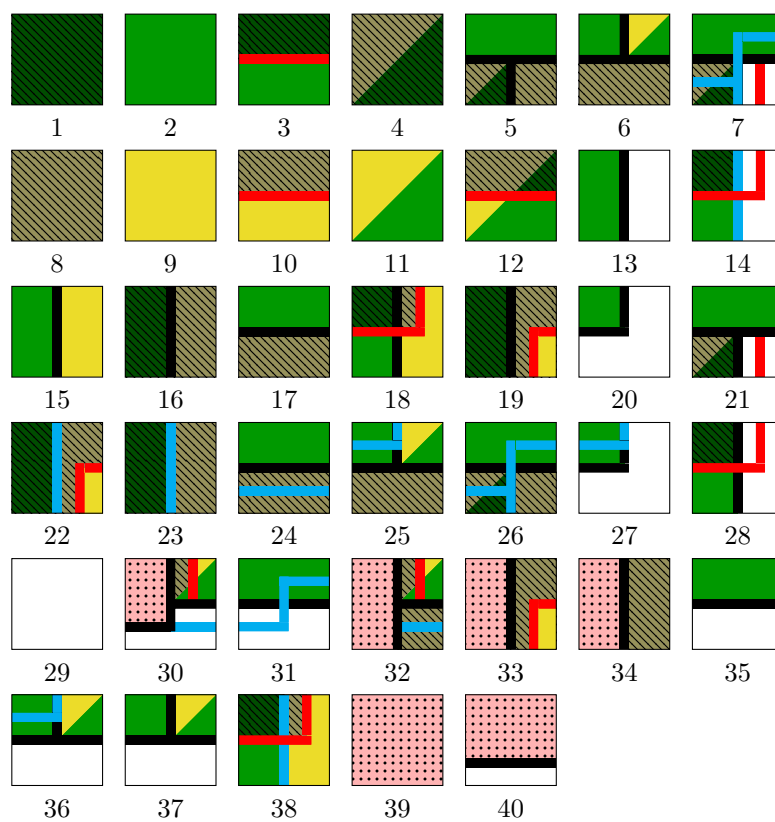
FIGURE 2.1 – Des grilles

Le moyen que nous avons trouvé afin d'éviter ce problème est de rendre la grille irrégulière, comme en 2.1c, où *a priori* il est possible de ne pas avoir le même problème.

Le jeu de tuiles de Wang T de la figure 2.2 va nous permettre d'obtenir le résultat. Remarquons tout d'abord que la configuration montrée en figure 2.3, que l'on appellera α , est valide. La configuration α forme de plus une grille dans laquelle on peut mettre du calcul, comme on le verra plus tard. Nous allons voir qu'il s'agit de la seule configuration qui contienne une grille infinie. Nous allons également expliquer comment mettre du calcul dedans.

Avant les preuves, définissons le vocabulaire permettant de décrire les points du SFT \mathcal{X}_T .

- La tuile 30 est la *tuile de coin*.
- Les tuiles 20 et 27 sont les *tuiles de fin*.
- Les tuiles 30, 32, 33, 34 sont des *tuiles de début*.
- Une *ligne horizontale* est un alignement horizontal connexe de tuiles contenant une ligne noire verticale (tuiles 5, 6, 7, 17, 21, 24, 25, 26, 31, 35, 36, 37), pouvant être fini à droite par des tuiles de fin, à gauche par des tuiles de début.
- Une *ligne verticale* est un alignement vertical connexe de tuiles contenant une ligne verticale noire ou bleue (tuiles 13, 14, 15, 16, 18, 19, 22, 23, 28, 32, 33, 34, 38) pouvant être terminé en haut par les tuiles 5, 21, 26, en bas par les tuiles 6, 20, 25, 27, 30, 36, 37.
- Une *diagonale* est une succession diagonale (positions $(i, j), (i + 1, j + 1), \dots$) de tuiles parmi 4, 11, 12.
- Un *carré* de taille k est un motif extensible de support $\llbracket 0, k + 1 \rrbracket^2$ tel que $\{0\} \times \llbracket 1, k \rrbracket$ et $\{k + 1\} \times \llbracket 1, k \rrbracket$ soient des lignes verticales et $\llbracket 1, k \rrbracket \times \{0\}$ et $\llbracket 1, k \rrbracket \times \{k + 1\}$ des lignes horizontales. Un carré n'a pas de ligne horizontale/verticale dans le sous-motif de support $\llbracket 1, k \rrbracket^2$. On peut remarquer que comme la couleur au dessus de la ligne horizontale du bas et celle du bas de la ligne horizontale du haut n'est pas la même, un carré contient nécessairement une diagonale aux positions (i, i) , pour $0 < i < k + 1$ ainsi qu'un signal de comptage quelque part entre les deux lignes horizontales.
- Un *signal de comptage* est un chemin connexe de tuiles parmi 3, 7, 10, 12, 14, 19, 22, 32, 33, 38, tel que le signal rouge soit continu. Un signal de comptage peut être terminé à gauche par les tuiles 30, 32 et à droite par les tuiles 7, 21. Le signal de comptage compte le nombre de carrés sur

FIGURE 2.2 – Le jeu de tuiles T .

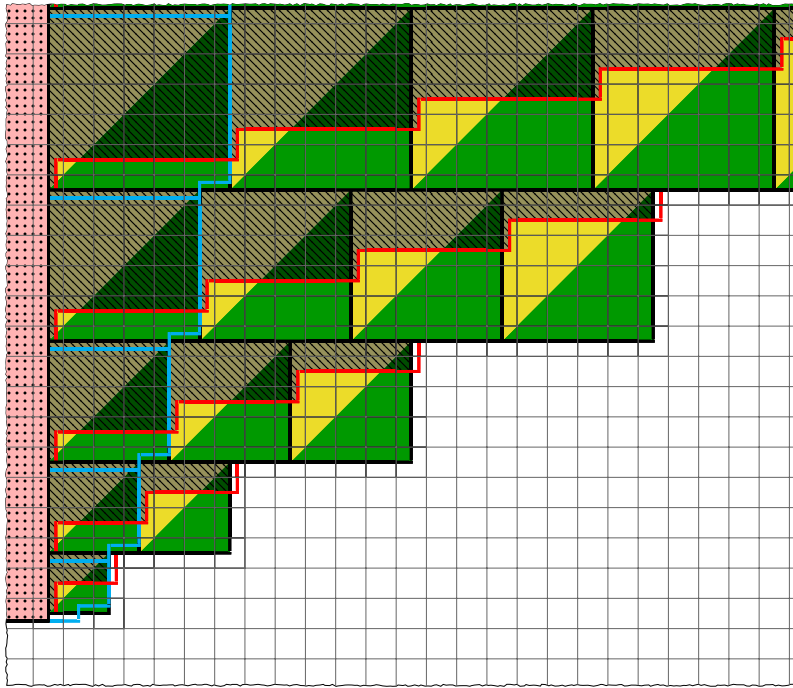


FIGURE 2.3 – Le point α , qui peut être vu comme une grille irrégulière lorsque l'on oublie tout sauf les lignes noires et le signal de croissance.

- chaque alignement horizontal et s'assure qu'il ne puisse pas y en avoir plus que la hauteur de ces carrés.
- Un *signal de croissance* est un chemin connexe de tuiles parmi 7, 14, 22, 23, 24, 25, 26, 27, 30, 31, 36, 38, tel que le signal bleu soit continu. Ce signal force les carrés à être plus grands de 1 sur la ligne du dessus.
 - Une *rangée* de carrés est un alignement horizontal de carrés.

Notons tout d'abord que lorsque la tuile de coin apparaît dans un point, alors celui-ci est nécessairement un translaté de α , le point de la figure 2.3 : en effet, au dessus de la tuile de coin apparaît nécessairement la tuile 33 et à sa droite la tuile 31 puis la tuile 27. Ces tuiles forcent l'existence d'un premier carré de côté 1 : la première rangée de carrés. Le signal de croissance force le premier carré de la rangée du dessus à être de côté 2 et le signal de croissance force celle-ci à avoir exactement 2 carrés. Et ainsi de suite...

Lemme 2.2.2

Le SFT \mathcal{X}_T admet une seule configuration, à translation près, contenant deux lignes horizontales ou plus : la configuration α de la figure 2.3.

Preuve. Soit x un point contenant deux lignes horizontales, ces deux lignes sont nécessairement face à face : soit elles sont infinies, soit elles finissent à gauche auquel cas elles finissent à la même colonne, les tuiles ne pouvant être que sur la

même colonne. On peut supposer sans perte de généralité qu'elles sont à distance $k + 1$ et qu'il n'y a pas de troisième ligne horizontale entre elles.

Comme les bords supérieurs et inférieurs d'une ligne horizontale ne sont pas de la même couleur, il doit nécessairement y avoir une ou plusieurs diagonales bien formées entre les deux lignes. Chaque diagonale force des lignes verticales, formant ainsi des carrés de taille k . De plus, ces carrés sont nécessairement coupés horizontalement par une ligne rouge, qui se qui se déplace verticalement de 1 à chaque fois qu'elle croise une ligne verticale, garantissant ainsi qu'il y ait exactement k carrés dans cette rangée, celle-ci n'est donc pas infinie.

Le signal de croissance apparaît nécessairement sur la ligne verticale de droite du carré le plus à gauche d'une rangée, et force l'existence de carrés de taille $k + 1$ dans la rangée du dessus et de taille $k - 1$ dans celle du dessous, forçant donc l'apparition de lignes horizontales. La tuile de coin apparaît forcément au début du carré de la rangée la plus basse, celle contenant exactement un carré, de taille 1. \square

On pourra noter que si la tuile de coin apparaît en $(0, 1)$, alors il y a des lignes horizontales de longueur $(k + 1)k + 1$ qui commencent à chaque position $(0, \frac{k(k+1)}{2})$: ces lignes sont le bas d'une rangée de k carrés. On notera que le signal de croissance forme une parabole.

Lemme 2.2.3

Le SFT \mathcal{X}_T est récursif : il existe une machine de Turing qui prenant en entrée un numéro de configuration et une coordonnée $z \in \mathbb{Z}^2$ donne le symbole en z de la configuration correspondante.

Preuve. D'après le lemme 2.2.2 il y a une seule configuration, à translation près, qui a deux lignes horizontales ou plus, les autres configurations sont alors de l'une des formes suivantes :

- Il y a une ligne horizontale, auquel cas le point peut avoir au plus une ligne verticale au dessus et une en dessous, sinon il y aurait forcément des carrés, et donc deux lignes horizontales. Il peut alors y avoir aussi un signal de comptage arbitrairement loin au dessus et en dessous. Toutes ces configurations sont descriptibles algorithmiquement.
- Il n'y a pas de ligne horizontale, il peut alors y avoir au plus une ligne verticale. Un signal de croissance peut là encore apparaître. Ces points sont également descriptibles algorithmiquement.

Tous les points de \mathcal{X}_T sont montrés en figures 2.3 et 2.4, ils sont tous descriptibles par une même machine de Turing. \square

Corollaire 2.2.4

Le SFT \mathcal{X}_T est dénombrable.

Il nous reste maintenant à voir comment insérer du calcul dans la configuration α afin de terminer la preuve du théorème 2.2.1. Remarquons tout d'abord qu'une seule ligne verticale termine sur une ligne horizontale du bas d'un carré :

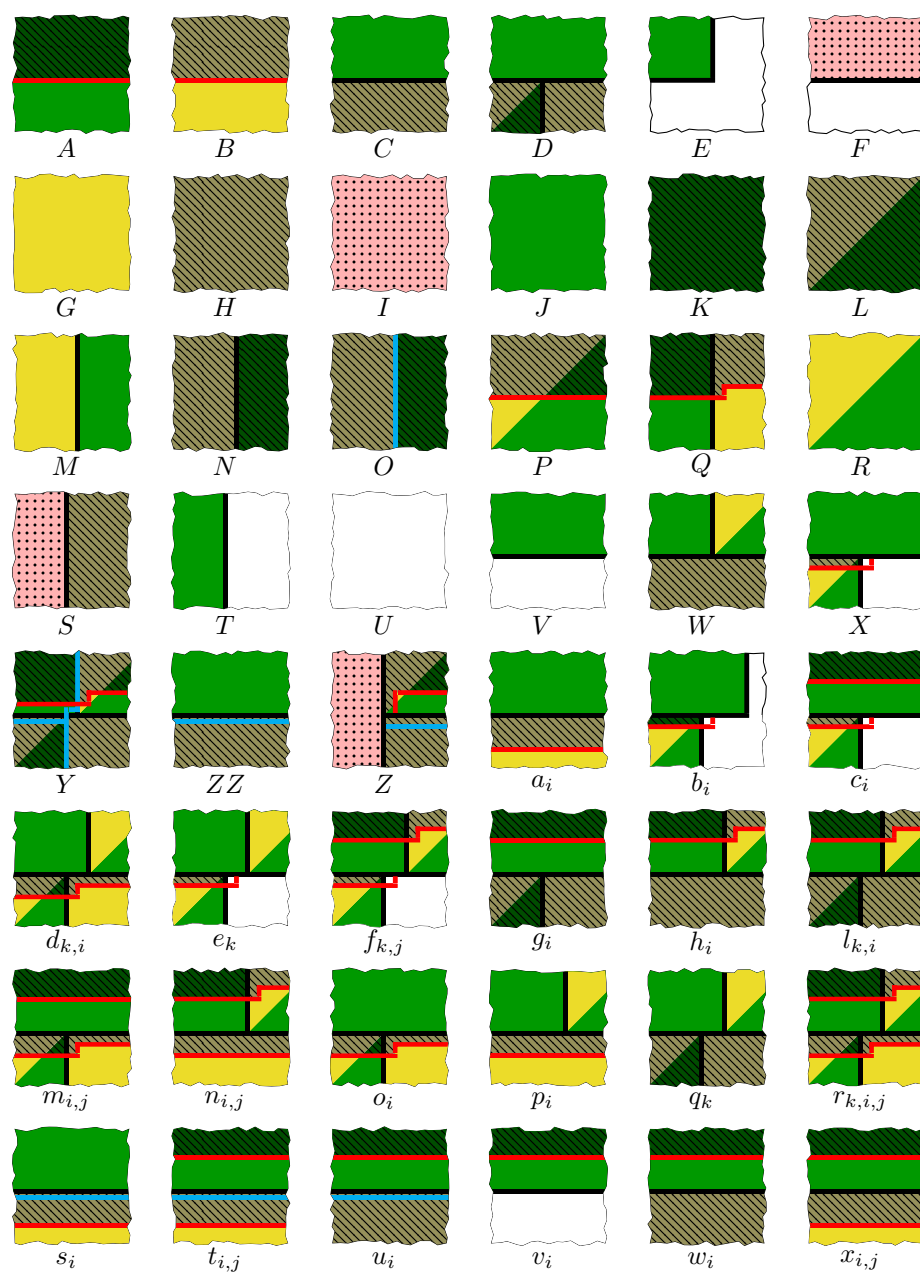


FIGURE 2.4 – Les configurations de $\mathcal{X}_T \setminus \{\alpha\}$, les indices correspondent au fait que les lignes verticales/horizontales et les signaux peuvent être à des distances différentes.

supposons que la tuile de coin soit d'abscisse 0, on voit alors que sur le k -ième alignement, contenant exactement k carrés de taille k , les lignes verticales sont aux abscisses $i(k+1)$, pour $0 \leq i \leq k$ et dans l'alignement du dessus, les lignes verticales sont aux coordonnées $i(k+2)$ avec $0 \leq i \leq k+1$. Comme

$$(i-1)(k+2) < i(k+1) < i(k+2) \quad \text{pour } 0 < i \leq k+1,$$

ceci est vrai pour tous les carrés sauf ceux le plus à gauche, pour lesquels la première ligne verticale est commune. On peut donc voir α comme une grille, dont le nombre d'intersections croît d'exactly 1 à chaque rangée de carrés.

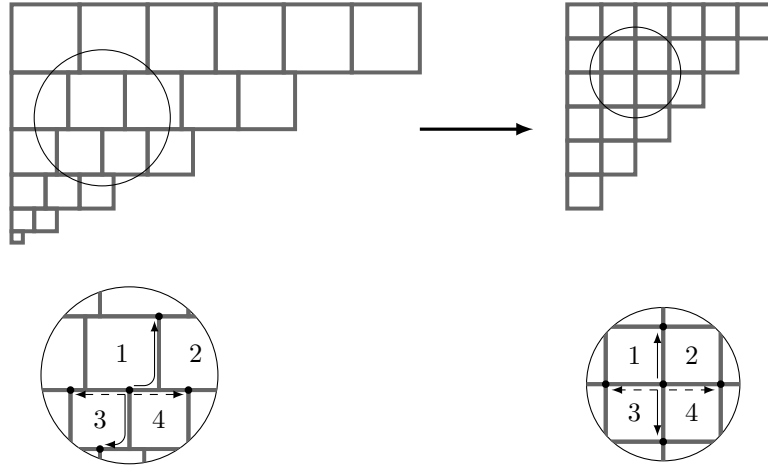


FIGURE 2.5 – Comment la grille de α peut être transformée en une grille régulière. En particulier, on peut voir comment une intersection peut transmettre de l'information à ses voisines.

Preuve du théorème 2.2.1. On peut donc coder une machine de Turing dans α , la grille croissant de 1 à chaque rangée, on a la place nécessaire pour le calcul. A chaque membre s de la classe $\Pi_1^0 S$, on peut donc associer un point alpha dont l'oracle est s . En quotientant par les translations de α , on obtient le résultat, voir la figure 2.5. \square

2.2.2 Degrés Turing des classes Π_1^0 et des SFTs

On s'intéresse maintenant aux ensembles de degrés Turing des classes Π_1^0 et à leur réalisation par des SFTs. Le premier résultat est un corollaire du théorème 2.2.1 et est le suivant :

Théorème 2.2.5

Pour toute classe $\Pi_1^0 S$, il existe un SFT X ayant les mêmes degrés Turing à l'ajout de $\mathbf{0}$ près, c'est à dire :

$$\{\deg_T s \mid s \in S\} \cup \{\mathbf{0}\} = \{\deg_T x \mid x \in X\}$$

Cela permet en particulier de résoudre le cas dénombrable : en effet une classe Π_1^0 dénombrable contient un membre récursif.

Corollaire 2.2.6

Pour toute classe Π_1^0 S contenant un membre récursif, il existe un SFT X ayant exactement le même ensemble de degrés Turing.

La question naturelle est maintenant de savoir si l'on peut aussi obtenir une caractérisation parfaite pour les classes Π_1^0 et les SFTs ne contenant pas de point récursif. Dans ce but, nous allons prouver le théorème ci-dessous, qui nous permet de comprendre mieux la structure calculatoire des sous-shifts.

Théorème 2.2.7

Jeandel et Vanier [JV12]

Tout sous-shift ne contenant pas de point récursif contient des points de degrés comparables. Plus exactement, pour tout sous-shift X , il existe un degré Turing \mathbf{d} , tel que X contient des points de tous degrés \mathbf{d}' tels que $\mathbf{d} \leq_T \mathbf{d}'$.

Cette structure n'est pas la même que celle des classes Π_1^0 , en effet on sait qu'il en existe certaines ayant des propriétés qu'aucun sous-shift ne peut avoir :

Théorème 2.2.1 (Jockusch et Soare [JS72]). *Il existe des classes Π_1^0 ne contenant aucun membre récursif et dont les membres sont deux à deux incomparables.*

On peut également déduire de cette contradiction que l'on ne peut pas faire mieux qu'un quasi-isomorphisme au théorème 2.2.1.

On va commencer par donner le résultat pour les sous-shifts de dimension 1, puis on généralisera aux dimensions supérieures, la preuve étant similaire mais légèrement plus technique.

Sous-shifts unidimensionnels

Un sous-shift ne contenant aucun point récursif est en particulier apériodique, et donc de cardinalité 2^{\aleph_0} , voir Durand [Dur99]. Une classe Π_1^0 sans point récursif a également comme cardinalité 2^{\aleph_0} , voir Cenzer et Remmel [CR98]. Étant donné une classe Π_1^0 S et un sous-shift X , tous deux sans points récursifs, il existe une injection de $S \rightarrow X$ et inversement, on souhaite ici rendre cette injection effective. Pour faire cela, on va utiliser le lemme suivant, qui va nous permettre de construire une injection effective d'une classe Π_1^0 S vers un

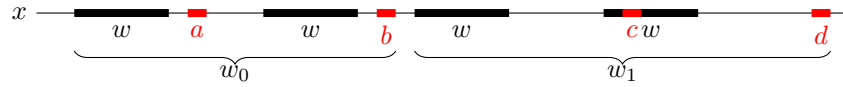


FIGURE 2.6 – Comment les occurrences les plus proches de w et la première lettre différente les suivant peuvent former les mots w_0 et w_1 . Quand on cherche les mots w_0, w_1 , les premiers que l'on trouve n'ont pas nécessairement la même longueur.

sous-shift minimal A sans point récursif.

Lemme 2.2.8

Soit x un point strictement quasipériodique d'un sous-shift minimal unidimensionnel A et \prec un ordre sur Σ_A . Pour tout mot w apparaissant dans x , il existe deux mots w_0 et w_1 apparaissant dans x tels que :

- w apparaît exactement deux fois dans w_0 et w_1 .
- Si a et b (resp. c et d) sont les premières lettres différentes dans les mots suivant chaque occurrence de w dans w_0 (resp. w_1), alors $a \prec b$ (resp. $d \prec c$).

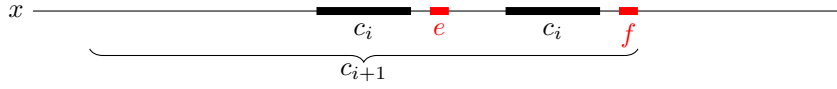
La figure 2.6 illustre w_0 et w_1 .

On rappelle qu'un sous-shift minimal est un sous-shift dont toutes les configurations ont exactement les mêmes motifs. Par exemple, si un point d'un sous-shift minimal admet un vecteur de périodicité, alors tous les autres points de ce sous-shift admettent le même vecteur de périodicité. Un sous-shift minimal ne contenant aucun point récursif ne peut donc avoir aucun point périodique.

Preuve. Par quasipériodicité de x , w apparaît une infinité de fois dans x . Comme x n'est pas périodique, deux occurrences distinctes de w ne peuvent pas être suivies de mots identiques. Soit y le mot le plus long apparaissant toujours à la suite de w dans x . Notons que w ne peut apparaître qu'une seule fois dans wy , sinon x serait périodique.

Par définition de y , les lettres après wy ne peuvent pas toujours être les mêmes. Il existe donc deux occurrences consécutives de wy avec des lettres a, b différentes qui leur succèdent. On peut supposer sans perte de généralité (l'autre cas étant similaire) que $a \prec b$. On définit alors w_0 comme le mot le plus court contenant les deux occurrences de wy et les lettres a, b qui les suivent.

x étant quasipériodique, une occurrence wyb apparaît aussi forcément avant une occurrence de wya , on peut trouver entre ces deux occurrences de wy deux occurrences de wy telles que les lettres leur succédant, c et d vérifient $d \prec c$. On définit alors w_1 comme précédemment. \square

FIGURE 2.7 – Construction des c_i .**Théorème 2.2.9**

Soit A un sous-shift unidimensionnel minimal contenant uniquement des points strictement quasipériodiques et x un point de A . Alors, pour tout degré Turing \mathbf{d} tel que $\deg_T x \leq \mathbf{d}$, il existe un point y de A de degré Turing \mathbf{d} .

Preuve. Nous allons montrer l'existence de deux fonctions calculables $f : A \times \{0, 1\}^{\mathbb{N}} \rightarrow A$ et $g : A \rightarrow \{0, 1\}^{\mathbb{N}}$ telles que pour tout $x \in A$ et $s \in \{0, 1\}^{\mathbb{N}}$, on aie $g(f(x, s)) = s$. En termes de degrés Turing, on a donc la relation suivante entre x et s :

$$\deg_T s \leq \deg_T f(x, s) \leq \sup_T(\deg_T x, \deg_T s) \quad (2.1)$$

On va donc donner deux algorithmes, l'un (f) qui à partir d'un point x de A et d'un élément s de $\{0, 1\}^{\mathbb{N}}$ construit un point y de A qui encode s de manière réversible, l'autre (g) qui décode s de y .

- Commençons par exhiber f . Soit \prec un ordre sur Σ_A . Étant donné un point $x \in A$ et un membre s de $\{0, 1\}^{\mathbb{N}}$, f va construire récursivement un autre point de A : l'algorithme commence avec un bloc $c_{-1} = x_0$ et construit inductivement des blocs c_i de plus en plus grands, et tels que c_{i+1} ait c_i en son centre. La séquence c_{-1}, c_1, c_2, \dots convergera donc vers un point y ayant chaque c_i en son centre. Il suffit de montrer comment on construit c_{i+1} à partir de c_i .

On cherche deux occurrences consécutives de c_i dans x , où les premières lettres différentes leur succédant a, b satisfont $a \prec b$ si $s_{i+1} = 0$ et $b \prec a$ si $s_{i+1} = 1$. D'après le lemme 2.2.8, on sait que l'on rencontrera forcément de telles occurrences. On définit maintenant c_{i+1} comme le bloc de x ayant la première des deux occurrences de w trouvées précédemment en son centre et contenant l'intégralité des deux occurrences et la première lettre différente succédant à chacune. La dernière lettre à droite de c_{i+1} est donc la lettre différente correspondant à la deuxième occurrence de w . On pourra se référer à la figure 2.7. On a donc construit la fonction f , qui est clairement récursive.

- On construit maintenant la fonction g : notons tout d'abord qu'à partir de y et c_i , on peut récupérer facilement c_{i+1} : il suffit de regarder à droite de l'occurrence de c_i apparaissant au centre de y et d'en trouver la première seconde occurrence, ceci nous donne le bloc c_{i+1} , et donc s_{i+1} également. Comme $y_0 = c_{-1}$, on peut reconstruire s de manière calculable.

Maintenant, si l'on prend un membre $s \in \{0, 1\}^{\mathbb{N}}$ tel que $\deg_T s > \deg_T x$, comme $\deg_T s = \sup_T(\deg_T x, \deg_T s)$, $y = f(x, s)$ a le même degré Turing que s d'après 2.1. \square

Le théorème 2.2.7, dans le cas de la dimension 1 est un corollaire de ce

théorème.

Sous-shifts de dimensions supérieures

Il faut maintenant généraliser le résultat aux dimensions supérieures. On ne peut pas réutiliser la méthode précédente directement, car il n'est pas vrai en dimension supérieure que toute configuration quasipériodique contient une ligne strictement quasipériodique. En effet, il existe des configurations strictement quasipériodiques, même dans des SFT apériodiques, contenant une ligne non quasipériodique (par exemple le jeu de tuiles apériodique de Robinson produit une configuration contenant une croix infinie), ou encore uniquement des lignes périodiques de périodes toutes différentes (de telles configurations apparaissent par exemple dans le SFT apériodique de Kari [Kar96]).

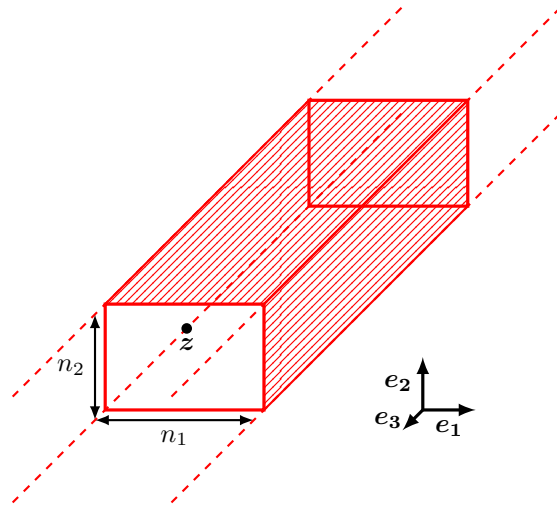


FIGURE 2.8 – Un faisceau d'une configuration d'un sous-shift de dimension 3. Sa direction est e_3 , de placement z .

Définition 2.2.10

Faisceau

Un *faisceau* de direction e_i d'une configuration $x \in \Sigma^{\mathbb{Z}^d}$ de dimension d est une fonction $l : \mathbb{Z} \times H \rightarrow \Sigma$, avec $H = z + \llbracket 0, n_1 \rrbracket \times \cdots \times \llbracket 0, n_{i-1} \rrbracket \times \{0\} \times \llbracket 0, n_{i+1} \rrbracket \times \cdots \times \llbracket 0, n_d \rrbracket$, avec $z \in \mathbb{Z}^d$ tel que $z_i = 0$ et $n_1, \dots, n_d \in \mathbb{N}$, telle que :

$$x|_{z+e_i\mathbb{Z}+H} = l$$

On appellera z le placement du faisceau et les n_i sa taille.

On peut voir un faisceau comme la restriction d'une configuration sur une droite épaisse, voir figure 2.8. L'ensemble des faisceaux d'une même taille H

forme un sous-shift de dimension 1 ayant comme alphabet Σ^H .

Lemme 2.2.11

Soit A un sous-shift minimal de dimension d et e_i une direction. Il existe un point $x \in A$ tel que tous les faisceaux de direction e_i de x soient quasipériodiques.

Remarque 2.2.1. *Toutes les configurations d'un SFT minimal n'ont pas toujours cette propriété. Dans un SFT minimal contenu dans celui de Robinson, il y a par exemple la croix infinie.*

Preuve. Soit $\{(z_k, n_k^1, \dots, n_k^d)\}_{k \in \mathbb{N}}$ une énumération de $\mathbb{Z}^d \times \mathbb{N}^{d-1}$, avec $z_{k_i} = 0$, et $H_k = z_k + \llbracket 0, n_k^1 \rrbracket \times \dots \times \llbracket 0, n_k^{i-1} \rrbracket \times \{0\} \times \llbracket 0, n_k^{i+1} \rrbracket \times \dots \times \llbracket 0, n_k^d \rrbracket$. Soit x une configuration, on note $p_k(x) : \mathbb{Z} \times H_k \rightarrow \Sigma$ la restriction de x à $e_i \mathbb{Z} + H_k$. On verra p_k comme une application de A dans $(\Sigma^{H_k})^{\mathbb{Z}}$, c'est à dire un sous-shift de dimension 1 d'alphabet Σ^{H_k} . Un *sous-shift e_i -directionnel* est un sous-ensemble de $\Sigma^{\mathbb{Z}^d}$ fermé, invariant par translations de direction e_i . Ce n'est pas un sous-shift au sens habituel du terme : c'est un sous-shift de dimension 1 sur un alphabet $\Sigma^{\mathbb{Z}^{d-1}}$.

On va construire inductivement des sous-shifts e_i -directionnels $A_k \subseteq A$ ayant la propriété que tout faisceau de direction e_i et de support H_j de n'importe quelle configuration de A_k , pour $j \leq k$, soit quasipériodique. Plus précisément, $p_j(A_k)$ sera un sous-shift minimal.

On va raisonner inductivement, on définit $A_{-1} = A$. Maintenant, supposons A_k défini, prenons $p_{k+1}(A_k)$. C'est un sous-shift non vide qui contient donc un sous-shift minimal X . On définit le sous-shift e_i -directionnel $A_{k+1} = p_{k+1}^{-1}(X) \cap A_k$. Par construction, $p_{k+1}(A_k)$ est minimal. De plus, pour tout $j < k$, $p_j(A_{k+1})$ est non vide et inclus dans $p_j(A_k)$ qui est minimal, $p_j(A_{k+1})$ est donc minimal.

Pour finir remarquons que, par compacité, $\bigcap_k A_k$ est non vide. \square

On peut maintenant prouver un analogue multidimensionnel du lemme 2.2.8 :

Lemme 2.2.12

Soit A un sous-shift minimal ne contenant que des points strictement quasipériodiques (ou de manière équivalente, un point strictement quasipériodique).

Soit \prec un ordre sur Σ_A . Il existe une direction e_i , telle que pour tout point $x \in A$ et pour tout bloc w apparaissant dans x , il existe deux blocs w_0 et w_1 apparaissant dans x tels que :

- w apparaisse exactement deux fois dans w_0 et w_1 sur la ligne de placement 0 de même taille que w .
- Les premières lettres différentes a, b entre les deux blocs contenant chaque occurrence de w en leur centre sont telles que $a \prec b$ dans w_0 et $b \prec a$ dans w_1 .

Le mot "premières" se réfère ici à une énumération adéquate de \mathbb{Z}^d .

Comme on l'a dit précédemment, un sous-shift minimal ne contenant aucun point récursif ne peut contenir aucun point périodique, il peut en revanche avoir des directions de périodicité en nombre strictement inférieur à la dimension.

Preuve. Remarquons tout d'abord, que comme A est strictement quasipériodique, aucune configuration ne peut être périodique. En particulier, il y a une direction e_i , telle qu'aucune configuration de A n'a de vecteur de périodicité colinéaire à e_i .

Ensuite, comme le résultat concerne l'existence d'un motif et que tous les points d'un sous-shift minimal ont les mêmes motifs, on peut se contenter de prouver l'existence d'un tel motif pour une configuration x dont tous les faisceaux de direction e_i sont quasipériodiques, qui existe d'après le lemme 2.2.11. Tous les faisceaux dans la suite seront de direction e_i .

Comme w apparaît dans x , il apparaît une seconde fois sur le faisceau le plus petit contenant w . Par non-périodicité de direction e_i , les deux occurrences de w sont au centre de blocs de même taille w', w'' différents.

On peut maintenant utiliser le même argument qu'au lemme 2.2.8 pour le faisceau contenant les deux blocs w', w'' . \square

Théorème 2.2.13

Soit A un sous-shift minimal dont aucun point n'est périodique et x un point de A . Alors pour tout degré Turing \mathbf{d} tel que $\deg_T x \leq \mathbf{d}$, il existe un point $y \in A$ de degré Turing \mathbf{d} .

Preuve. La preuve est quasiment identique au cas unidimensionnel, utilisant cette fois-ci le lemme 2.2.12, la seule différence étant, pour la fonction f , qu'il faut chercher c_i en parallèle sur tous les faisceaux de direction e_i afin de trouver c_{i+1} . En revanche, les fonctions f, g ne sont plus les mêmes pour tous les sous-shifts, étant donné qu'il faut trouver la direction e_i selon laquelle les configurations de A ne sont pas périodiques. On pourra se référer à la figure 2.9 pour un exemple de comment c_{i+1} est construit à partir de c_i . \square

Corollaire 2.2.14

Tout sous-shift non vide X ne contenant que des points non récursifs a des points de degrés Turing différents mais comparables.

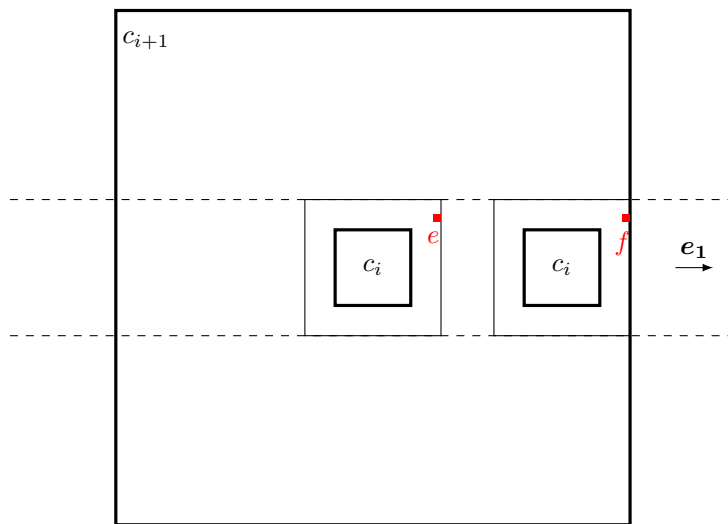


FIGURE 2.9 – Construction de c_{i+1} à partir de c_i dans le cas de la dimension 2 où e_1 n'est pas un vecteur de périodicité.

Chapitre 3

Caractérisation des ensembles de périodes des SFTs

Sommaire

3.1	Diverses notions de périodicité	55
3.2	Périodes horizontales	57
3.3	Fortes périodes	62
3.4	Nombre de points fortement périodiques	69
3.5	1-périodes	72
3.6	Ensembles de Périodes des sous-shifts sofiqes et effectifs	77
3.7	Remarques sur les périodes en tant que langages	80

En section 1.1.7 on a défini la notion de périodicité, un point étant périodique si son orbite $\{\sigma_z(x) \mid z \in \mathbb{Z}^d\}$ est finie.

En dimension d , cela signifie avoir d vecteurs de périodicité non colinéaires, ou de manière équivalente l'existence de $n \in \mathbb{N}$ tel que $x = \sigma_{ne_i}(x)$, pour tout $1 \leq i \leq d$. Se limiter à cette définition de périodicité serait cependant réducteur, étant donné qu'une configuration ayant un vecteur de périodicité admet tout de même une certaine forme de périodicité. Pour cela, on introduit plusieurs notions de périodicité, pour un point x , on note $\Gamma_x = \{\mathbf{v} \in \mathbb{Z}^d \mid \forall \mathbf{z} \in \mathbb{Z}^d, x_{\mathbf{z}+\mathbf{v}} = x_{\mathbf{z}}\}$ le treillis des vecteurs de périodicité de x :

- x est *fortement périodique* de période $n > 0$ si $\Gamma_x = n\mathbb{Z}^d$, n est la forte période de x ,
- x est *1-périodique* s'il existe 1 vecteur $\mathbf{v} \in \mathbb{N} \times \mathbb{Z} \setminus \{\mathbf{0}\}$ tel que $\Gamma_x = \mathbf{v}\mathbb{Z}$, \mathbf{v} est la 1-période de x
- x est *périodique horizontalement* s'il existe $n \in \mathbb{N}^*$ tel que $n\mathbf{e}_1\mathbb{Z} \subseteq \Gamma_x$, la période horizontale est le plus petit tel n .

Les ensembles de périodes fortes/1-périodes/périodes horizontales sont des invariants de conjugaison. Dans le cas des périodes fortes, on peut même compter le nombre de points pour chaque forte période, on obtient alors une suite p_n qui est invariante par conjugaison.

En dimension 1, toutes les notions de périodicité évoquées plus haut sont équivalentes et le nombre de points périodiques est donné par $\text{tr } A^k$, où A est

une matrice représentant le SFT. Une caractérisation des nombres de points périodiques possibles est donnée par Kim, Ormes et Roush [KOR00].

Dans ce chapitre nous allons caractériser les différents ensembles de périodes à l'aide de classes de complexité. L'idée vient du fait que l'on peut coder les calculs d'une machine de Turing fonctionnant en temps t et en espace s dans un rectangle de taille $s \times t$ avec des tuiles de Wang.

Dans un premier temps, en section 3.1, nous discuterons des diverses notions de périodicité et de quelques-unes de leur propriétés. Les autres sections sont dévolues à la caractérisation de ces notions de périodicité :

- En section 3.2, nous prouverons que les ensembles de périodes horizontales en dimension 2 sont exactement les langages L , tels que $L \in \mathbf{NSPACE}(2^n)$.
- En section 3.3, nous prouverons que les ensembles de périodes fortes correspondent exactement aux langages L tels que $L \in \mathbf{NE}$.
- En section 3.4, nous caractériserons les fonctions $f_X : \mathbb{N} \rightarrow \mathbb{N}$ qui à n associe le nombre de points de forte période n de X en montrant que ce sont exactement les fonctions de $\#\mathbf{E}$.
- En section 3.5, nous nous intéresserons aux ensembles de $(d-1)$ -périodes en montrant que ce sont aussi les langages L tels que $L \in \mathbf{NSPACE}(2^n)$.

Habituellement, lorsque l'on dit qu'un langage L est dans une classe de complexité, cela signifie que $\text{bin}(L)$, le langage codé en binaire, est dans cette classe de complexité. Le codage de l'entrée a de l'importance : par exemple la classe \mathbf{NP} sur des langages unaires n'est pas la même que sur les langages binaires, elle correspond à la classe \mathbf{NE} sur langages binaires ($\text{bin}(L) \in \mathbf{NE} \Leftrightarrow \text{un}(L) \in \mathbf{NP}$). Nous n'utiliserons ici que des astuces et résultats classiques de complexité que l'on pourra par exemple trouver dans les livres de Balcazar, Diaz et Gabarro [BDG88, BDG90]. Les résultats de cette partie ont été partiellement publiés dans [JV10a, JV10b].

3.1 Diverses notions de périodicité

Étant donné un point x , on note $\Gamma_x = \{v \in \mathbb{Z}^d \mid \forall z \in \mathbb{Z}^d, x_{z+v} = x_z\}$ le treillis des vecteurs de périodicité de x .

Définition 3.1.1

Périodicités

Γ_x peut prendre plusieurs formes différentes :

- $\Gamma_x = \{\mathbf{0}\}$, alors x n'a aucun vecteur de périodicité et est donc aperiodique.
- Γ_x est de rang d , la dimension de l'espace, alors l'orbite de x est finie et x est *périodique*. Dans ce cas, il existe $n \in \mathbb{N}^*$ tel que $n\mathbb{Z}^d \subseteq \Gamma_x$ on appelle *période* le plus petit tel n . Quand on a en particulier $n\mathbb{Z}^d = \Gamma_x$, on dit alors que x est *fortement périodique* de *forte période* n .
- Γ_x est de rang intermédiaire $1 \leq k \leq d$, x est *k-périodique*.

De manière équivalente, il existe alors k vecteurs indépendants v_1, \dots, v_k tels que $\Gamma_x = v_1\mathbb{Z} + \dots + v_k\mathbb{Z}$

En dimension 2, on dira que x est *périodique horizontalement* de période horizontale n , quand $n \in \mathbb{N}^*$ est le plus petit entier tel que $n\mathbb{Z} \times \{0\} \subseteq \Gamma_x$.

Pour un sous-shift X , on notera \mathfrak{P}_X , \mathfrak{P}_X^1 et \mathfrak{P}_X^h les ensembles de fortes périodes, 1-périodes et périodes horizontales de X respectivement. Ce sont tous les trois des invariants de conjugaison.

Lemme 3.1.2

Étant donné un SFT $X \subseteq \Sigma^{\mathbb{Z}^d}$ de dimension d , et un vecteur $v = (v_1, \dots, v_d) \in \mathbb{Z}^d \setminus \{\mathbf{0}\}$, il existe un SFT $(d-1)$ -dimensionnel $X' \subseteq (\Sigma^{F_v})^{d-1}$, où $F_v = \llbracket 0, v_1 \rrbracket \times \dots \times \llbracket 0, v_d \rrbracket$ tel qu'il existe un isomorphisme entre X' et les configurations périodiques selon v de X .

Preuve. On introduit une fonction de groupage, $\Gamma : \Sigma^{\mathbb{Z}^d} \rightarrow (\Sigma^{F_v})^{\mathbb{Z}^d}$ définie de la manière suivante : pour tout $x \in \Sigma^{\mathbb{Z}^d}$, si $y = \Gamma(x)$, alors :

$$\forall p \in F_v \text{ et } \forall z = (z_1, \dots, z_d) \in \mathbb{Z}^d, \text{ on note } z' = ((v_1 + 1)z_1, \dots, (v_d + 1)z_d), \\ \text{on a } x_{z'+p} = y_z(p).$$

Notons $Y = \Gamma(X)$, Y est encore un SFT. Les configurations périodiques selon v de X correspondent aux configurations périodiques selon v' de Y , où $v'_i = 1$ si $v_i \neq 0$ et 0 sinon.

Maintenant, comme $v \in \mathbb{Z}^d \setminus \{\mathbf{0}\}$, il existe $1 \leq i \leq d$ tel que $v'_i \neq 0$, soit

$$\pi : (\Sigma^{F_v})^{\mathbb{Z}^d} \rightarrow (\Sigma^{F_v})^{\mathbb{Z}^{d-1}}$$

la projection vers l'hyperplan généré par $(e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_d)$ et $\mathbf{0}$. Comme

$$v'\mathbb{Z} + \sum_{j \neq i} e_j\mathbb{Z} = \mathbb{Z}^d,$$

les points périodiques selon v' de Y sont entièrement générés par leur projection. Enfin, remarquons que les projections de points périodiques selon v' de Y sont définissables par une famille de motifs interdits : les contraintes selon la dimension qui a disparu sont remplacées par des contraintes supplémentaires sur les autres dimensions. \square

	$x(1,11)$	$x(2,11)$	$x(0,9)$	$x(1,9)$	$x(2,9)$	$x(0,7)$	$x(1,7)$	$x(2,7)$	$x(0,5)$	$x(1,5)$	$x(2,5)$	$x(0,3)$	$x(1,3)$	$x(2,3)$
11	$x(1,11)$	$x(2,11)$	$x(0,9)$	$x(1,9)$	$x(2,9)$	$x(0,7)$	$x(1,7)$	$x(2,7)$	$x(0,5)$	$x(1,5)$	$x(2,5)$	$x(0,3)$	$x(1,3)$	$x(2,3)$
10	$x(1,10)$	$x(2,10)$	$x(0,8)$	$x(1,8)$	$x(2,8)$	$x(0,6)$	$x(1,6)$	$x(2,6)$	$x(0,4)$	$x(1,4)$	$x(2,4)$	$x(0,2)$	$x(1,2)$	$x(2,2)$
9	$x(1,9)$	$x(2,9)$	$x(0,7)$	$x(1,7)$	$x(2,7)$	$x(0,5)$	$x(1,5)$	$x(2,5)$	$x(0,3)$	$x(1,3)$	$x(2,3)$	$x(0,1)$	$x(1,1)$	$x(2,1)$
8	$x(1,8)$	$x(2,8)$	$x(0,6)$	$x(1,6)$	$x(2,6)$	$x(0,4)$	$x(1,4)$	$x(2,4)$	$x(0,2)$	$x(1,2)$	$x(2,2)$	$x(0,0)$	$x(1,0)$	$x(2,0)$
7	$x(1,7)$	$x(2,7)$	$x(0,5)$	$x(1,5)$	$x(2,5)$	$x(0,3)$	$x(1,3)$	$x(2,3)$	$x(0,1)$	$x(1,1)$	$x(2,1)$	$x(0,-1)$	$x(1,-1)$	$x(2,-1)$
6	$x(1,6)$	$x(2,6)$	$x(0,4)$	$x(1,4)$	$x(2,4)$	$x(0,2)$	$x(1,2)$	$x(2,2)$	$x(0,0)$	$x(1,0)$	$x(2,0)$	$x(0,-2)$	$x(1,-2)$	$x(2,-2)$
5	$x(1,5)$	$x(2,5)$	$x(0,3)$	$x(1,3)$	$x(2,3)$	$x(0,1)$	$x(1,1)$	$x(2,1)$	$x(0,-1)$	$x(1,-1)$	$x(2,-1)$	$x(0,-3)$	$x(1,-3)$	$x(2,-3)$
4	$x(1,4)$	$x(2,4)$	$x(0,2)$	$x(1,2)$	$x(2,2)$	$x(0,0)$	$x(1,0)$	$x(2,0)$	$x(0,-2)$	$x(1,-2)$	$x(2,-2)$	$x(0,-4)$	$x(1,-4)$	$x(2,-4)$
3	$x(1,3)$	$x(2,3)$	$x(0,1)$	$x(1,1)$	$x(2,1)$	$x(0,-1)$	$x(1,-1)$	$x(2,-1)$	$x(0,-3)$	$x(1,-3)$	$x(2,-3)$	$x(0,-5)$	$x(1,-5)$	$x(2,-5)$
2	$x(1,2)$	$x(2,2)$	$x(0,0)$	$x(1,0)$	$x(2,0)$	$x(0,-2)$	$x(1,-2)$	$x(2,-2)$	$x(0,-4)$	$x(1,-4)$	$x(2,-4)$	$x(0,-6)$	$x(1,-6)$	$x(2,-6)$
1	$x(1,1)$	$x(2,1)$	$x(0,-1)$	$x(1,-1)$	$x(2,-1)$	$x(0,-3)$	$x(1,-3)$	$x(2,-3)$	$x(0,-5)$	$x(1,-5)$	$x(2,-5)$	$x(0,-7)$	$x(1,-7)$	$x(2,-7)$

FIGURE 3.1 – Comment les configurations périodiques selon $v = (3, 2)$ d'un SFT bidimensionnel de rayon $r = 1$ peuvent être transformées en un SFT unidimensionnel. Toutes les colonnes de cellules groupées sont identiques (à décalage vertical près), le groupage a été délimité par les lignes en gras. Le SFT 1d est constitué d'une colonne de cellules groupées. Le voisinage initial, bidimensionnel, montré à gauche, peut être transformé en un nouveau voisinage, unidimensionnel, montré à droite : les cellules groupées voisines à gauche et à droite sont également au dessus et en dessous dans la colonne.

Corollaire 3.1.3

Soit X un SFT de dimension d et k vecteurs v_1, \dots, v_k de \mathbb{Z}^d non colinéaires, il existe un SFT X' de dimension $(d - k)$ tel que X' soit isomorphe aux configurations périodiques selon v_1, \dots, v_k de X .

Remarque 3.1.1. On peut trouver cet isomorphisme de manière récursive.

3.2 Périodes horizontales

Dans cette section, nous étudions les SFTs 2-dimensionnels et leurs ensembles de périodes horizontales. Ce cas, qui ne paraît pas naturel d'un point de vue dynamique, nous permet d'introduire en douceur la technique de preuve qui sera utilisée par la suite.

Lemme 3.2.1

Si X est un SFT 2-dimensionnel, alors $un(\mathfrak{P}_X^h) \in \mathbf{NSPACE}(n)$.

Preuve. Soit $X = \mathcal{X}_{\mathcal{F}}$ un SFT de dimension 2 sur l'alphabet Σ . On construit une machine de Turing non déterministe acceptant 1^p si et seulement si $p + 1$ est une période horizontale de X . La machine doit fonctionner en espace $\mathcal{O}(p)$, l'entrée étant donnée en unaire.

Soi r le rayon de \mathcal{F} , un point x appartient à X si ses motifs de support \mathcal{B}_r ne contiennent aucun motif de \mathcal{F} . On l'a vu précédemment, s'il existe un point de période horizontale p , alors il en existe un de période horizontale p et de période verticale au plus $|\Sigma|^{(2r+1)p}$.

- On initialise un tableau P de taille p tel que $P[i] = 1$ pour tout i , qui servira à vérifier que p est la plus petite période horizontale, ceci nécessite un espace $\mathcal{O}(p)$
- On choisit non déterministiquement $v \leq |\Sigma|^{(2r+1)p}$ (la période verticale) que l'on code en binaire (dont la taille est donc plus petite que p).
- On choisit non déterministiquement $2r$ lignes horizontales $(l_i)_{0 \leq i \leq r-1}$ de période p . C'est à dire que l'on choisit $2r \times p$ symboles de Σ .
- Pour tout $2r < i \leq v$, on choisit non déterministiquement une ligne horizontale de période p (on choisit donc p symboles) et on vérifie qu'aucun motif de support \mathcal{B}_r dans les lignes l_i, \dots, l_{i-2r} ne contienne un motif interdit. A chaque étape, il suffit de garder en mémoire les $2r$ -dernières lignes devinées ainsi que les $2r$ premières lignes
- A chaque étape précédente, on vérifie si la ligne devinée est périodique de période k pour chaque $k < p$, si ce n'est pas le cas, alors $P[k] \leftarrow 0$.
- Pour $i \leq 2r$, on vérifie que $l_{v-i}, \dots, l_v, l_0, \dots, l_{i-2r}$ ne contiennent aucun motif interdit.
- On rejette s'il existe k tel que $P[k] = 1$, on accepte sinon.

Cet algorithme nécessite de conserver $4r$ lignes et P et v en mémoire à chaque étape, et fonctionne donc en espace $\mathcal{O}(p)$. \square

La véritable difficulté se situe dans la réciproque, il va nous falloir construire un SFT dont les périodes horizontales sont exactement les entiers codés en unaire acceptés par une machine fonctionnant en temps linéaire.

Lemme 3.2.2

Soit $L \subseteq \mathbb{N}^*$ un langage, si $un(L) \in \mathbf{NSPACE}(n)$, alors il existe un SFT X de dimension 2 tel que $L = \mathfrak{P}_X^h$.

Preuve. Soit M une machine de Turing acceptant $un(L)$ en espace linéaire : sur une entrée 1^n , on peut supposer que la machine n'utilisera jamais plus de $n + 1$ cases. On sait qu'il existe une constante c , telle que pour une entrée de taille n la machine s'arrête en exactement c^n étapes. On va construire un SFT X' tel que $n \in L$ si et seulement si $n + 4$ est une période horizontale d'un point de X' . La preuve va s'effectuer en deux temps :

- On exhibe d'abord un SFT Y_c tel que tous les points de période horizontale n de Y_c ressemblent à une grille formée de rectangles de taille $n \times c^{n-1}$ délimités par des marqueurs horizontaux et verticaux, voir figure 3.2. Les périodes horizontales de Y_c sont $\mathbb{N} \setminus \{0, 1\}$. On construit ce SFT dans le lemme 3.2.3.
- On code la machine de Turing M dans ces rectangles à l'aide du jeu de tuiles de la figure 3.3. Celle-ci étant non déterministe, on doit faire attention à bien synchroniser les transitions non déterministes pour tous les rectangles voisins horizontalement, de manière à ce que la période reste la largeur de ces rectangles.
Étant donné qu'à chaque étape de temps il ne peut y avoir qu'une seule transition non-déterministe, il suffit pour cela de rajouter une couche T , périodique horizontalement, dont les symboles représentent toutes les transitions possibles, la transition en cours dans un rectangle doit alors être la même que celle sur la ligne de T qui la recouvre. Les transitions non déterministes sont alors les mêmes pour tous les rectangles alignés horizontalement.

La difficulté réside dans la première partie, nous la prouvons au lemme 3.2.3.

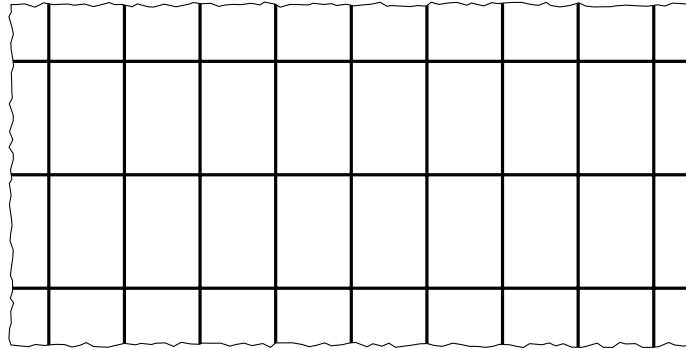


FIGURE 3.2 – Un point périodique du SFT Y_c , la base de la construction. Un point horizontalement périodique de Y_c a nécessairement la structure ci-dessus : une grille de rectangles, dont la largeur est exactement la période n et dont la hauteur est c^{n-1} .

□

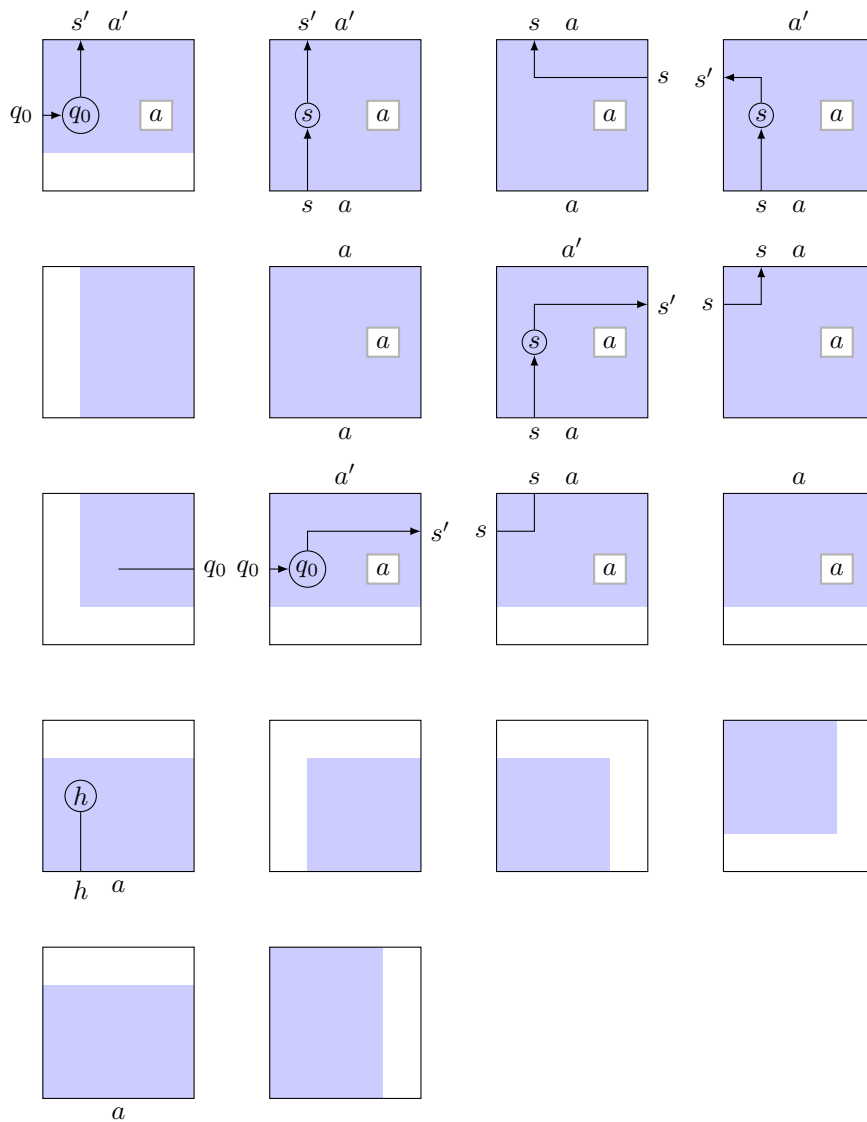


FIGURE 3.3 – Nouveau jeu de tuiles pour coder les machines de Turing : on a rajouté les tuiles des deux lignes du bas. Les nouvelles tuiles permettent d'accepter des calculs qui terminent. Un rectangle à bords blancs forme un calcul valide, sa hauteur correspond au temps mis par la machine à s'arrêter et sa largeur à l'espace utilisé.

Lemme 3.2.3

Pour tout entier $c \geq 2$, il existe un SFT Y_c tel que tout point $y \in Y_c$ de période horizontale p est formé de rectangles de taille $p \times c^{p-1}$ dont les bords sont distingués. De plus tout entier $p \geq 2$ est une période horizontale.

Preuve. On construit le SFT Y_c en superposant plusieurs composantes A, C_c, T , ayant les buts suivants :

- A force les points périodiques à avoir des lignes verticales marquées.
- C_c force l'apparition de lignes horizontales dans les points périodiques, formant ainsi des rectangles avec les lignes verticales. Cette composante forcera en même temps tous les rectangles à être de la même taille.
- Avec T , on force exactement une ligne horizontale à apparaître dans chaque période horizontale.

On donne maintenant ces composantes et les règles qui les régissent :

- La première composante, A , est composée d'un SFT W apériodique, Est-déterministe¹, dont on appellera les symboles *blancs*, et d'un symbole \blacksquare . L'alphabet de A est donc $\Sigma_A = \Sigma_W \cup \{\blacksquare\}$. On conserve les règles sur les symboles de W et on ajoute les règles suivantes :

- il ne peut pas y avoir de symbole blanc au dessus ou en dessous d'un \blacksquare ,
- deux \blacksquare ne peuvent pas apparaître côte à côte horizontalement.

Un point périodique de A de période p doit nécessairement avoir des lignes verticales de \blacksquare séparant des colonnes de symboles blancs, étant donné que W est un SFT apériodique.

Pour le moment, rien n'interdit plus d'une ligne de \blacksquare d'apparaître dans une période, ni même les lignes de \blacksquare d'être à des distances différentes. La figure 3.4 montre une forme possible d'un point périodique à cette étape.



FIGURE 3.4 – Un point périodique de A , les symboles $w_{i,j}$ sont des blancs.

- La deuxième couche de symboles $C_c = P_c \times \{\blacksquare, _ \}$ va produire des lignes horizontales de manière à ce que les points de période np puissent être constitués de rectangles de taille $p \times c^{p-1}$ délimités par les symboles \blacksquare et $_$. L'idée est la suivante : on considère chaque ligne horizontale entre deux lignes de \blacksquare comme un mot de l'alphabet $P'_c = \{0, \dots, k-1\} \times \{0, 1\}$, c'est à

1. On utilisera et expliquera cette propriété en temps utile.

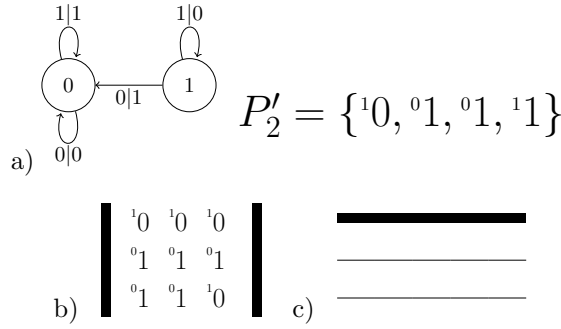


FIGURE 3.5 – Le transducteur correspondant à $c = 2$ et les symboles du SFT correspondant. Un motif valide de C_2 en (b) et (c) : le **I** en (b) vient de la composante A et correspond au symbole 1 de la composante C_2 .

dire comme un nombre a entre 0 et $c^{p-1} - 1$ écrit en base c . On s'assure ensuite avec des contraintes locales que le mot de la ligne suivante est $a + 1 \pmod{c^{p-1}}$. La composante $\{0, 1\}$ sert de retenue. On ajoute une retenue 1 qu'on superposera uniquement au symbole **I**, afin d'obtenir P_c . On peut voir le transducteur permettant de coder les contraintes locales dans le cas $c = 2$ sur la figure 3.5.

Maintenant, avec $\{\blacksquare, _ \}$, on marque les lignes correspondant au nombre 0, afin de marquer une ligne sur les c^{p-1} . Cette ligne est la seule telle où un symbole $^0 1$ à droite de **I** a un symbole $^1 0$ au dessus de lui.

Enfin, on interdit un $_$ d'apparaître à gauche ou à droite d'un **■**. Cela force les compteurs de chaque colonne de symboles blancs à être mis à zéro en même temps, et donc à avoir les mêmes hauteur et largeur.

La figure 3.6a montre un pavage possible à cette étape : la période horizontale n'est pas nécessairement la distance entre les lignes verticales. Elle peut être plus grande car les symboles blancs entre deux lignes verticales ne sont pas nécessairement les mêmes.

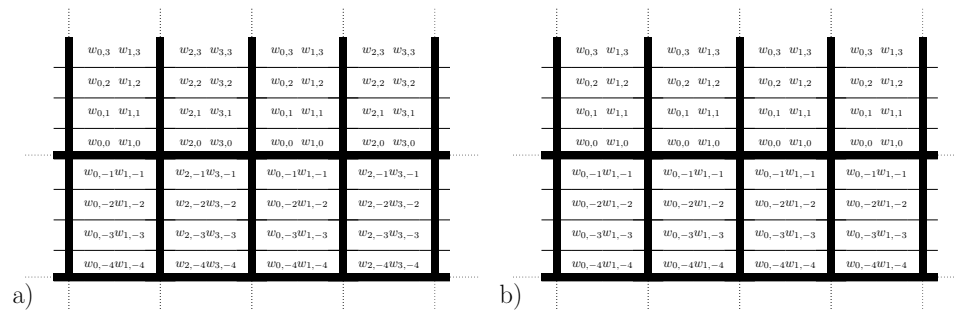


FIGURE 3.6 – En (a), un exemple de point périodique après la superposition de C_2 à A : la distance entre deux lignes verticales consécutives est la même partout, mais n'est pas nécessairement la période. En (b), après ajout de la composante T , la largeur des rectangles est exactement la période horizontale. On ne représente ici que la composante A et les symboles $\blacksquare, _$ de C_2 .

- La dernière composante T va permettre de palier à cela. L'alphabet de T est

exactement le même que celui de W de la composante A . On va se servir du fait que W est Est-déterministe. Un SFT *Est-déterministe* a la propriété suivante : étant donnés deux symboles a, b aux positions (i, j) et $(i, j + 1)$, alors il a au plus un symbole possible en position $(i + 1, j)$. Cela signifie en particulier que si l'on se donne une colonne de symboles, alors tout le demi-plan à sa droite est déterminé par celle-ci². Sur cette couche, on force les symboles à deux positions adjacentes horizontalement d'être les mêmes : chaque ligne horizontale est constituée d'un unique symbole.

On force également le symbole sur T à droite d'un \blacksquare à être le même que celui de la composante W . Cela signifie que tous les symboles à droite d'un \blacksquare sur une même ligne horizontale sont les mêmes et donc en particulier que toutes les lignes verticales de symboles de W à droite de lignes verticales de \blacksquare sont les mêmes. Comme W est Est-déterministe, les colonnes de symboles blancs entre deux lignes verticales de \blacksquare sont donc les mêmes.

A ce stade, on a fini la construction de Y_c : les points périodiques sont nécessairement formés de rectangles dont la largeur est la période, comme en figure 3.6b.

Vérifions que le SFT obtenu est bien comme on le souhaite : soit $y \in Y_c$ un point horizontalement périodique de période horizontale p :

- La composante A force l'apparition de lignes verticales de \blacksquare au minimum une fois dans la période.
- La composante C_c force l'apparition de rectangles de dont la largeur divise p .
- La composante T force la période horizontale à être la largeur des rectangles.

□

La preuve est finie et on a prouvé le théorème suivant, caractérisant les ensembles de périodes horizontales des SFTs de dimension 2 :

Théorème 3.2.4

Soit $L \subseteq \mathbb{N}^*$ un langage, les trois propositions suivantes sont équivalentes :

- Il existe un SFT X tel que $L = \mathfrak{P}_X^h$.
- $un(L) \in \mathbf{NSPACE}(n)$
- $bin(L) \in \mathbf{NSPACE}(2^n)$

3.3 Fortes périodes

On s'intéresse maintenant aux fortes périodes, nous allons prouver une caractérisation des ensembles de fortes périodes des SFTs. Néanmoins, on sera contraint de considérer les SFTs de toutes dimensions pour obtenir une caractérisation intéressante. Ceci est dû au fait que pour une dimension d donnée, il y a un écart entre la complexité de tester si un entier est une forte période et la place disponible pour coder du calcul dans une configuration fortement périodique :

². Cela ne signifie pas nécessairement que celui-ci existe, mais s'il existe, il est entièrement déterminé par la colonne.

- Vérifier que n est une forte période nécessite un temps n^d .
- Une configuration de forte période n forme des d -cubes de côté n , il y a donc une place n^d pour coder du calcul.

Le problème est que l'on ne sait coder un calcul d'une machine de Turing terminant en temps n^d que dans un "espace" de taille au moins n^{2d} . Par exemple, dans un carré de taille $n \times n$, on ne peut pas mettre un calcul terminant en plus de n étapes de temps, et vérifier un qu'un tel carré ne contient pas de motif interdit nécessite n^2 étapes de temps. On aurait donc besoin d'un SFT de dimension $2d$ pour coder un calcul de $\mathbf{NTIME}(n^d)$, mais vérifier qu'un entier est une forte période d'un SFT de dimension $2d$ nécessitera une machine fonctionnant en temps $\mathbf{NTIME}(n^{2d})$, et ainsi de suite.

On a donc un écart que l'on avait pas pour la caractérisation des périodes horizontales. Cet écart n'est pas surprenant dans la mesure où les classes de complexité en temps sont moins robustes aux changements de modèle de calcul que les classes de complexité en espace.

En considérant les SFTs de toutes dimensions à la fois, cela revient à considérer la classe $\mathbf{NP} = \bigcup_{d \in \mathbb{N}} \mathbf{NTIME}(n^d)$. Dans cette section, nous montrons donc la caractérisation suivante :

Théorème 3.3.1*Caractérisation des périodes*

$L \subseteq \mathbb{N}$ est l'ensemble des fortes périodes d'un SFT si et seulement si $un(L) \in \mathbf{NP}$.

Comme pour le résultat sur les périodes horizontales, la preuve aura lieu en deux temps, le sens de la réalisation d'un langage de \mathbf{NP} en tant que périodes d'un SFT étant le plus compliqué. Notons ici que l'on parle en termes de classes de complexité sur des langages unaires, en complexité classique, la classe correspondante est $\mathbf{NE} = \bigcup_{c \in \mathbb{N}} \mathbf{NTIME}(2^{cn})$.

Lemme 3.3.2

Soit X un SFT de dimension d , alors $un(\mathfrak{P}_X) \in \mathbf{NP}$.

Preuve. Il suffit de prendre la machine qui fait la chose suivante : sur l'entrée p , elle devine non-déterministiquement un cube n^d , il suffit ensuite de vérifier que ce cube ne contient aucun motif interdit (cela prend $\mathcal{O}(n^d)$ étapes) et que n est bien la période et donc en particulier vérifier que pour tout $k < n$, k n'est pas une période (cela prend $\mathcal{O}(n^{2d})$ étapes). \square

Pour l'autre direction, on va procéder comme précédemment : on construit d'abord des SFTs Y_d de dimension d , tels que les points périodiques de Y_d soient des cubes de dimension d (que l'on appellera d -cubes) dont les arêtes sont marquées. Dans un deuxième temps, on codera le calcul des machine de Turing

dans ce SFT, obtenant ainsi la réalisation voulue.

Lemme 3.3.3

Pour toute dimension d , il existe un SFT d -dimensionnel Y_d tel que :

- Tout point périodique est fortement périodique.
- Tout point périodique $y \in Y_d$ est formé de d -cubes p^d , où p est la période de y .
- Tout entier $p \geq 2$ est une période, et donc une période forte.

Preuve. La construction sera à nouveau basée sur un jeu de tuiles apériodique, dont on cassera l'apériodicité à l'aide de tuiles supplémentaires pour forcer une structure régulière. Y_d sera donc constitué de trois couches A, S, T :

- La composante A forcera les points périodiques à avoir des lignes.
- La composante S forcera les points périodiques à être constitués de cubes.
- La composante T forcera les points périodiques à être constitués de cubes dont la longueur des côtés est la période.


On donne maintenant ces composantes dans le détail.

- Pour construire le SFT A , on part d'un SFT 2-dimensionnel apériodique NO-déterministe quelconque³. À nouveau, nous donnerons la définition de NO-déterminisme au moment où nous l'utiliserons. On définit un SFT A' de dimension 2 :

- L'alphabet de A' est l'alphabet de W (dont les éléments sont les *blancs*) incrémenté de trois symboles : **l**, **+** et **–**.
- On conserve les motifs interdits de W et on ajoute les motifs interdits suivants :
 - * Au dessus et en dessous de **l**, il ne peut y avoir que **l** ou **+**.
 - * A gauche et à droite de **–**, il ne peut y avoir que **–** ou **+**.
 - * A gauche ou à droite de **+**, il ne peut y avoir que **–**.
 - * Au dessus et en dessous de **+**, il ne peut y avoir que **l**.

Les points périodiques de A' contiennent nécessairement des symboles **l**, **–** et/ou **+**. C'est à dire qu'il y a : soit une infinité de lignes de **–**, soit une infinité de lignes de **l**, soit une infinité de rectangles dont les côtés sont formés par des **–, l** et dont les coins sont des **+**.

On obtient A à partir de A' en gardant le même alphabet, ainsi que les mêmes règles sur les deux premières dimensions et en forçant les symboles voisins selon les autres dimensions à être identiques. On appellera *plan* A' le plan de A sur qui a conservé les règles de A' .

- La deuxième couche S va forcer des carrés à apparaître sur le plan A' dans les points périodiques et créera les frontières des d -cubes sur toutes les dimensions. Pour chaque $2 \leq i \leq d$, on définit S_i , un SFT sur l'alphabet  :

- Les règles pour S_i sur le plan défini par e_1, e_i sont des règles de Wang : les bords de chaque symbole doivent correspondre avec ceux de ses voisins, les règles gauche droite correspondent à $\pm e_i$ et les règles haut/bas à $\pm e_1$.

3. Pour ce théorème cela ne change rien, en revanche celui de Kari [Kar92] a une propriété intéressante qui sera exploitée dans la section suivante.

- Les règles pour les autres dimensions sont que s'il y a un symbole a à la position $z \in \mathbb{Z}^d$, alors il y a également un symbole a aux positions $z \pm e_k$, pour $k \neq 1, i$.

On va superposer les SFTs S_i afin d'obtenir S : à chaque coordonnée, les symboles sur tous les S_i doivent être d'un seul des deux ensembles $\{\text{■}, \text{■}\}$ ou $\{\text{■}, \text{■}, \text{■}, \text{■}\}$. La figure 3.7 montre comment la composante S_2 est superposée sur le plan A' , forçant les configurations périodiques à être formées de carrés sur cette couche. La figure 3.8 montre comment les composantes S_i sont superposés entre elles.

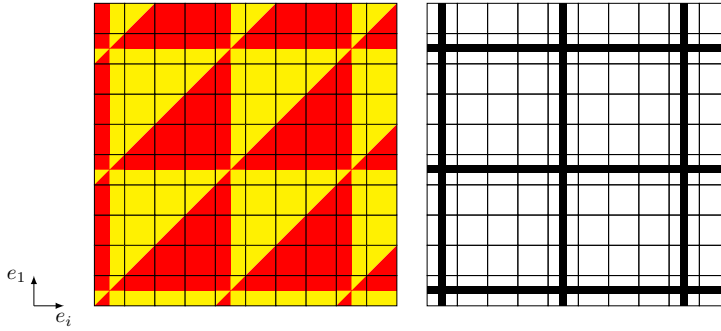


FIGURE 3.7 – À gauche la composante S_2 et à droite la composante A' : la superposition des deux force l'existence de carrés sur ce plan dans les points périodiques.

Enfin, on donne les règles de superposition de S sur la couche A : on impose des règles uniquement par rapport à la couche S_2 (qui fait des carrés sur le plan A'). On ne peut superposer un ■ qu'à un ■ , un ■ qu'à un ■ et un ■ qu'à un ■ , les autres symboles sont forcés superposés sur des blancs.

Les points fortement périodiques à ce stade sont nécessairement constitués de d -cubes dont les coins sont marqués par des $\text{■}, \dots, \text{■}$ et dont les faces sont marquées par les ■ et ■ : si les d -cubes sont de côté n , et qu'il y a un coin en $\mathbf{p} = (p_1, \dots, p_d) \in \mathbb{Z}^d$, alors pour toute coordonnée $\mathbf{q} = (q_1, \dots, q_d) \in \mathbb{Z}^d$ il y a un ■ ou un ■ sur la composante S_k si et seulement si $\mathbf{p}_k \equiv \mathbf{q}_k \pmod{n}$ et il y a un ■ ou un ■ si et seulement si $\mathbf{p}_1 \equiv \mathbf{q}_1 \pmod{n}$. Ceci est dû au fait que sur le plan A' , les lignes ne peuvent pas avoir la même couleur de chaque côté sur la composante S_2 : il y a donc nécessairement des carrés (à cause de la diagonale) dans un point périodique. Les carrés formés sur S_2 forcent l'apparition de carrés de même taille sur les autres composantes S_k .

A ce stade, un point périodique sera donc nécessairement constitué de d -cubes, cependant le côté de ces cubes n'est pas nécessairement la période, car sur le plan A' , les carrés formés par S_2 ne contiennent pas nécessairement les mêmes symboles blancs à l'intérieur. La période est donc un multiple du côté des d -cubes.

- La dernière étape consiste donc à synchroniser les symboles blanc à l'intérieur de chaque carré sur le plan A' . On va utiliser pour cela le NO-déterminisme du SFT apériodique W à la base de A' : Un SFT est *Nord-Ouest-déterministe* (NO-déterministe) s'il a la propriété suivante : étant donnés deux symboles a, b aux positions (i, j) et $(i+1, j+1)$ alors il y a au plus un symbole possible en position $(i+1, j)$. Il suffit donc de transmettre les symboles de la ligne

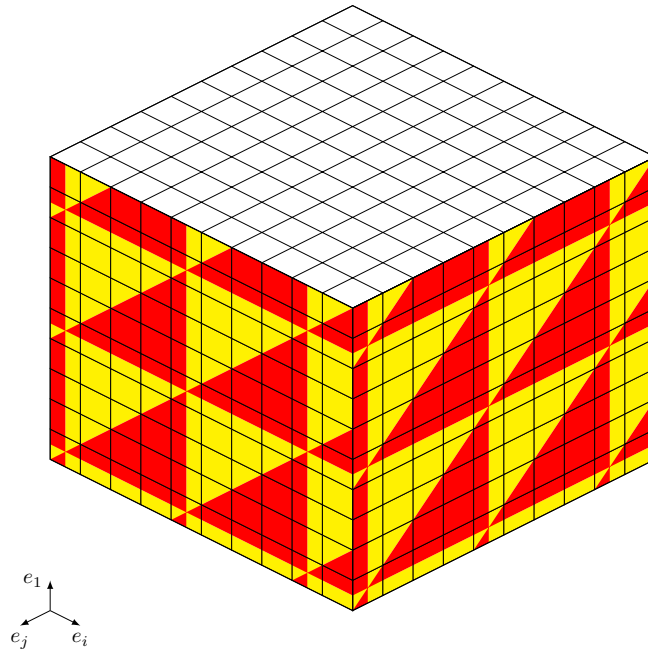


FIGURE 3.8 – Comment les composantes S_i et S_j sont superposées l'une à l'autre et forment des cubes : sur le plan e_1, e_i on voit la composante S_i et sur le plan e_1, e_j , on voit la composante S_j .

horizontale la plus haute et la ligne verticale la plus à gauche de chaque carré vers leurs voisins, comme en figure 3.9 pour synchroniser les arrières plans de tous les carrés. On utilise pour cela 4 sous-composantes V, V', H, H' dont les alphabets sont pour chacune les symboles blancs de A . Ces composantes agiront uniquement sur le plan A' , et donc les symboles sur chacune d'entre elles seront les mêmes aux positions $z \in \mathbb{Z}^d$ et $z \pm e_j$, avec $3 \leq j \leq d$. Il suffit donc de donner les règles sur le plan A' :

- V sert à synchroniser la ligne la plus à gauche d'un carré avec celle des carrés à gauche et de droite. Les règles sont simples, le symbole en $z \in \mathbb{Z}^d$ sur la couche V est le même que ceux à sa droite et à sa gauche, les positions $z \pm e_2$. La seule règle de superposition est que lorsque l'on est à droite d'un symbole \blacksquare , alors le symbole sur V est le même que sur A' .
- V' sert à synchroniser la ligne la plus à gauche d'un carré avec celle des carrés en haut à droite et en bas à gauche : le symbole en position $z \in \mathbb{Z}^d$ sur V' est le même que celui en position $z \pm (e_1 + e_2)$. Pour la superposition, comme avant, lorsque l'on est à droite d'un \blacksquare , alors le symbole sur V' est le même que celui sur A' .
- H sert à synchroniser la ligne la plus haute d'un carré avec celle des carrés en haut et en bas. Le symbole en $z \in \mathbb{Z}^d$ sur la couche H est le même que ceux juste en haut et juste en bas, c'est à dire aux positions $z \pm e_1$. La règle de superposition est que lorsque l'on est en dessous d'un \blacksquare , alors le symbole sur A' est le même que celui sur H .
- H' sert à synchroniser la ligne la plus haute d'un carré avec celle des carrés en haut et en bas. Le symbole en $z \in \mathbb{Z}^d$ sur la couche H' est le même que

ceux juste en haut et juste en bas, c'est à dire aux positions $z \pm (e_1 + e_2)$. La règle de superposition est la même que pour H .

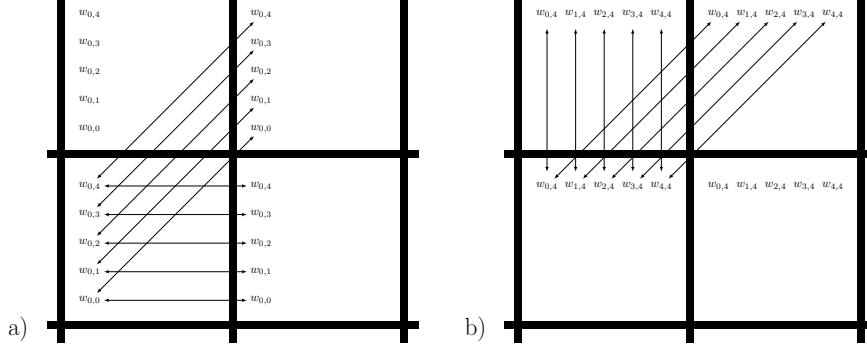


FIGURE 3.9 – En (a) la manière dont V et V' synchronisent les colonnes les plus à gauche de tous les carrés et en (b) comment H et H' synchronisent les lignes les plus hautes de tous les carrés.

La construction est maintenant terminée. Si on prend un point périodique de Y_d de période p . Par construction il est formé d' d -cubes de côté p dont les coins sont marqués par $(\blacksquare, \dots, \blacksquare)$ et dont les arêtes sont marquées par les symboles $(\blacksquare, \dots, \blacksquare)$ et $(\blacksquare, \dots, \blacksquare)$. \square

Lemme 3.3.4

Soit $L \subseteq \mathbb{N}$ un langage tel que $un(L) \in \mathbf{NTIME}(n^d)$, il existe alors un SFT $2d$ -dimensionnel X tel que $p \in L$ si et seulement s'il existe un point $c \in X$ de forte période p et tel que tout point périodique soit fortement périodique.

Preuve. Soit M une machine reconnaissant L . Il faut construire un SFT X_M dont les fortes périodes sont exactement les entiers acceptés par M . On va utiliser le SFT Y_{2d} construit au lemme 3.3.3, il suffit de montrer comment on met un calcul terminant en temps n^d dans un cube n^{2d} . On a donc exactement l'espace nécessaire dans un $2d$ -cube pour coder un tel calcul.

L'idée est de plier le diagramme espace-temps de la machine, de manière à ce qu'il tienne dans le $2d$ -cube tout en gardant des contraintes locales pour le coder, ceci a déjà été fait par Borchert [Bor08] et Jones et Selman [JS74] par exemple.

Dans un diagramme espace temps de la machine M sur l'entrée n , les cellules ont des coordonnées (t, s) , avec $t \leq n^d, s \leq n^d$, t représentant l'étape de temps et s la position sur le ruban. On doit maintenant transformer toute cellule (t, s) en une cellule d'un $2d$ -cube de manière à ce que deux positions consécutives du diagramme espace-temps soient voisines. On transforme donc t, s en vecteurs de $\llbracket 0, n-1 \rrbracket^d$ ((t, s) est donc un vecteur de $\llbracket 0, n-1 \rrbracket^{2d}$) à l'aide d'un code réfléchi en base n , appelé aussi code de Gray, voir Knuth [Knu05].

La position $(t_0, \dots, t_{d-1}) \in \llbracket 0, n-1 \rrbracket^d$ représente l'entier $t = \sum a_i n^i$, où $a_i = t_i$ si la somme $\sum_{j>i} t_j$ est paire et $a_i = n-1-t_i$ sinon. La position suivante

est donc donnée par la somme des parités des positions de poids plus fort. Pour transformer ce codage en contraintes locales, il suffit d'avoir une couche de parité P .

La couche de parité est P est faite de multiples sous-couches $P_i = \{0, 1\}$, une pour chaque direction e_i , $2 \leq i \leq d$. On donne les règles de la sous-couche P_i , on est dans un d -cube aux arêtes et coins marqués. On a un coin en position $z_c = (z_{c1}, \dots, z_{cd}) \in \mathbb{Z}^d$, il a forcément 0 sur la couche P_i . Les règles sont maintenant les suivantes s'il y a b en position $z \in \mathbb{Z}^d$, alors il y a $b + 1 \pmod 2$ en position $z + e_i$ et b aux positions $z + e_j$ pour $j \neq i$. Il n'y a pas de règles lorsque l'on passe d'un d -cube à un autre. On peut voir en figure 3.10 comment le codage de parité se fait dans un 3-cube.

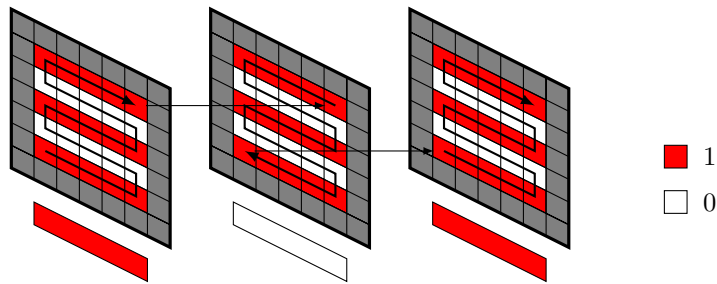


FIGURE 3.10 – Comment les couches de parité sont codées dans un 3-cube et l'ordre des cases déterminé par celles-ci : la parité donne la direction dans laquelle est la prochaine case.

On a donc deux couches de parité : une pour l'espace, sur les d premières dimensions et une sur les d suivantes pour le temps.

On a maintenant vu comment coder une machine de Turing M dont les calculs terminent en temps n^d sur le SFT Y_{2d} . Il ne reste plus qu'à nous assurer que l'on peut bien synchroniser les transitions non-déterministes entre tous les $2d$ -cubes.

Pour faire cela, on ajoute une composante N de synchronisation des transitions non déterministes constituée de sous-couches dont les alphabets sont à chaque fois l'ensemble des transitions possibles de M :

- La première sous-couche \sqsupset va permettre de savoir sur toute cellule à une étape de temps quelle transition a lieu : Sur une cellule effectuant une transition, la composante \sqsupset doit contenir la même transition. Si il y a un symbole l sur \sqsupset en position z , alors il doit y avoir le même symbole en position $z \pm e_i$, pour $1 \leq i \leq d$, c'est à dire les dimensions correspondant à l'espace.
- On a ensuite un ensemble de sous-couches \sqsubset_i , une pour chacune des dimensions correspondant au temps, $d + 1 \leq i \leq 2d$. La composante \sqsubset_i est régie par les règles suivantes : le symbole sur \sqsubset_i en position z est le même que celui en position $z \pm (e_i + e_1)$. Le symbole sur la couche \sqsubset_i est le même que celui sur la composante \sqsupset quand on est sur le bord du cube sur la composante 1. Dans la construction du lemme 3.3.3, cela correspond à quand la composante S_i contient un \blacksquare ou un \blacklozenge . La figure 3.11 montre comment cette synchronisation s'effectue. En dimension 2, cela correspondrait à des composantes se comportant de manière similaire aux composantes V, V' de la construction précédente.

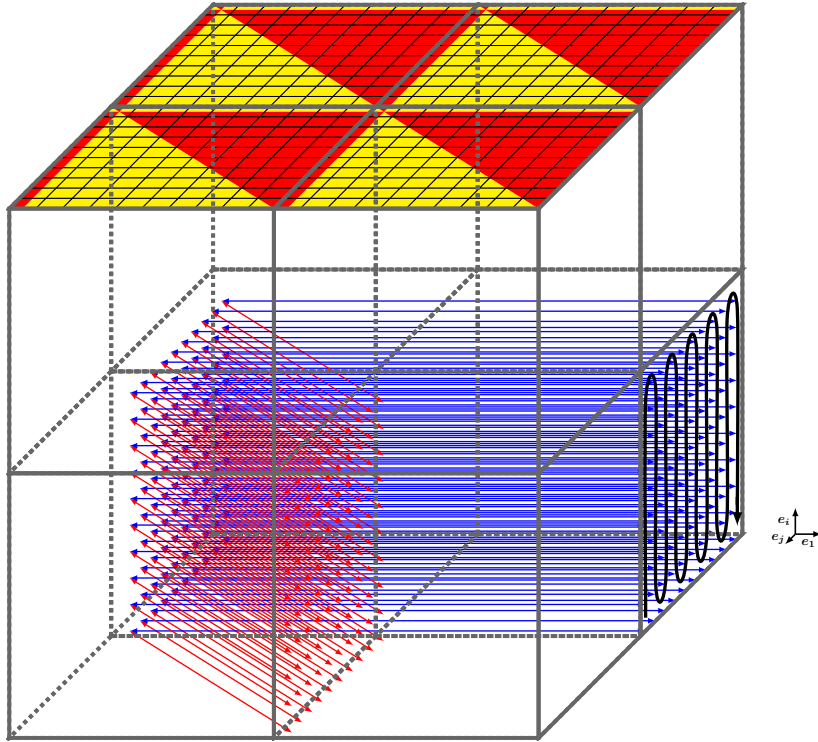


FIGURE 3.11 – La synchronisation des transitions non-déterministes entre les différents $2d$ -cubes : on a ici la projection sur trois dimensions, e_1, e_i, e_j . La composante \sqcap est représentée en bleu et \sqsupset_j est représentée en rouge. Les dimensions e_i, e_j sont des dimensions correspondant au temps et e_1 est une dimension correspondant à l'espace. Sur le haut de la figure, on a représenté la composante S_j de la construction du lemme 3.3.3. Les couches \sqcap et \sqsupset_j se synchronisent lorsque l'on est sur le bord d'un carré sur S_j .

Comme précédemment, $n + 4$ est une (forte) période si et seulement si n est accepté par M , pour obtenir exactement n , on doit à nouveau "épaissir" les symboles sur les bordures.

□

3.4 Nombre de points fortement périodiques

Dans le théorème 3.3.1 de la section précédente, nous avons vu que les ensembles \mathfrak{P}_X des SFTs correspondent exactement aux langages de la classe **NE**. Dans cette section, on va plus loin en comptant le nombre de points périodiques : on va caractériser la suite $(p_n(X))_{n \in \mathbb{N}^*}$ où $p_n(X)$ est le nombre de points de X de forte période n . De manière intuitive, les fonctions qui comptent le nombre de points périodiques sont à mettre en relation avec les fonctions qui comptent le nombre de chemins acceptants d'une machine de Turing de la classe **NE**, c'est

à dire les fonctions de la classe $\#E$.

Théorème 3.4.1

Pour tout SFT X de dimension d , soit $\mathfrak{N}_X(n)$ la fonction

$$n \mapsto p_n(X)/n^d.$$

L'ensemble des fonctions $\mathfrak{N}_X(n)$ pour tous les SFTs de toutes dimensions est exactement $\#E$.

La division par n^d est due à une renormalisation : en effet, chaque point de forte période n a exactement n^d translatés.

Preuve. Il y a deux sens à cette preuve :

- On a montré au lemme 3.3.2 que pour un SFT X , vérifier si $n \in \mathfrak{P}_X$ est dans **NP** pour n codé en unaire. Le nombre de chemins acceptants de la machine décrite dans cette preuve est exactement le nombre de points de période n . Pour normaliser par n^d , il suffit à chaque fois que l'on devine un cube n^d de vérifier si l'on est plus petit, pour l'ordre lexicographique, que ses translatés par chaque v , pour $\|v\|_\infty \leq n$.
- On a réalisé n'importe quelle fonction de **NP** pour une entrée codée en unaire en tant que périodes d'un SFT dans le lemme 3.3.4. Il faut modifier cette preuve afin que le nombre de points de période n soient exactement le nombre de chemins acceptants de la machine de Turing. Le SFT construit avait une structure particulière formée de $2d$ -cubes créés par Y_{2d} : dans Y_{2d} il y avait autant de points avec un coin de $2d$ -cube à l'origine⁴ que de fonds blancs apériodiques dans les carrés de la composante A' . Il suffit donc de forcer pour une taille de cube le fond apériodique à toujours être le même.

On peut faire cela en prenant le SFT apériodique NO-déterministe de Kari [Kar92]. Ce SFT est exactement le même que celui de Robinson [Rob71], avec une composante supplémentaire constituée de flèches, voir la figure 3.12. On peut voir en figure 3.14 un carré valide par ce SFT dont les bords hauts et gauche sont triviaux, et peuvent donc être forcés à l'aide de règles locales.

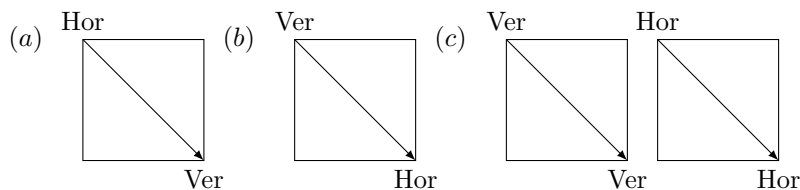


FIGURE 3.12 – Les flèches ajoutées par Kari au SFT de Robinson afin de le rendre NO-déterministe. Les marquages des flèches doivent correspondre à leurs extrémités. Sur les bras horizontaux on superpose le symbole (a) et sur les verticaux (b). Les symboles (c) et (d) peuvent être superposés aux croix uniquement.

□

4. Forcer le fait d'avoir un coin à l'origine correspond au niveau du comptage à la renormalisation par n^{2d} .

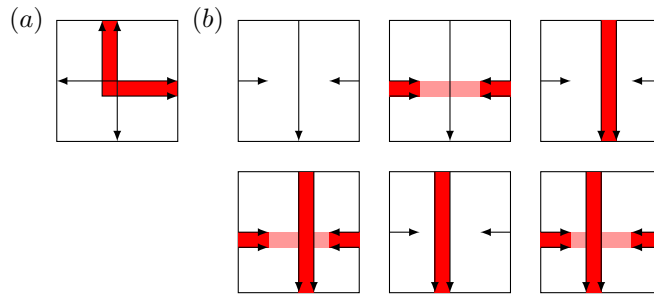


FIGURE 3.13 – Les tuiles de Robinson : en (a) les croix et en (b) les bras.

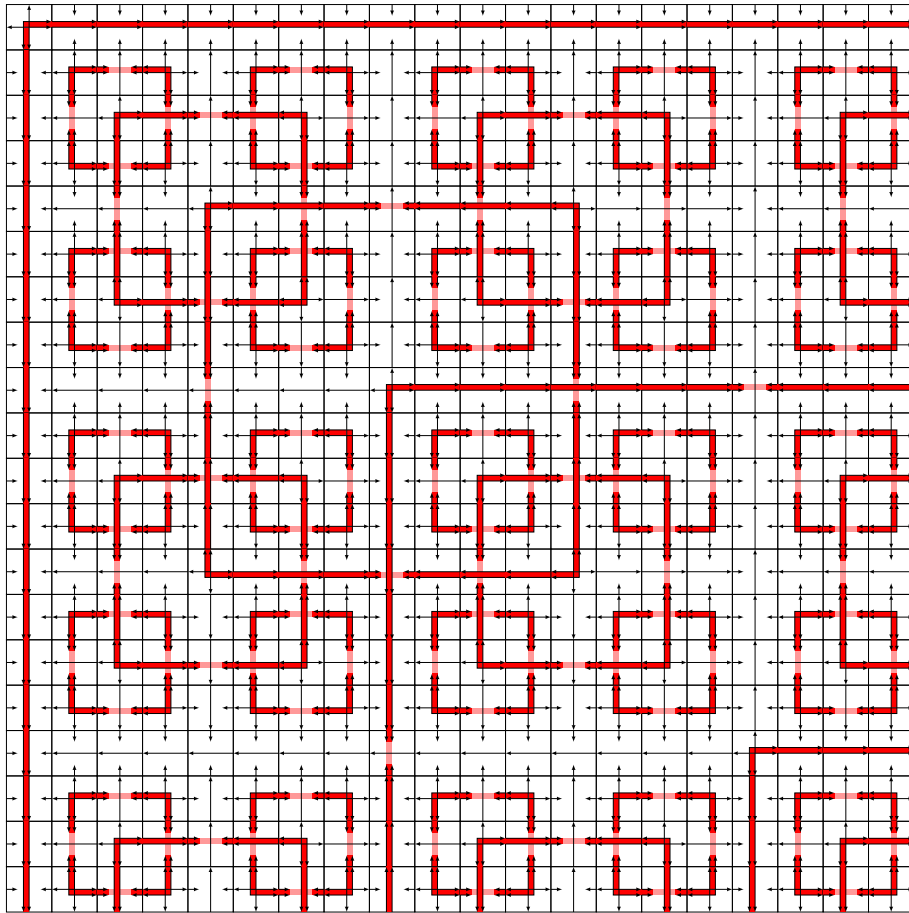


FIGURE 3.14 – Un carré valide avec le SFT de Kari : les flèches ne sont pas représentées, mais sont aisément déductibles. La ligne horizontale la plus haute et la ligne verticale la plus à gauche déterminent tout le carré. Notons que ces deux lignes sont très faciles à générer, et qu'elles permettent de générer des carrés de taille arbitrairement grande.

3.5 1-périodes

On s'intéresse maintenant à nouveau aux SFTs de dimension 2, et en particulier aux ensembles de 1-périodes qu'ils peuvent avoir. Une configuration 1-périodique est une configuration qui n'a qu'une seule direction de périodicité. On va prouver ici que les ensembles de 1-périodes, des ensembles de vecteurs de \mathbb{Z}^2 , sont exactement les ensembles de vecteurs de \mathbb{Z}^2 pouvant être reconnus par une machine en espace non déterministe linéaire pour une entrée donnée en unaire. Le résultat, ainsi que sa preuve, ressemblent beaucoup au théorème 3.2.4 : on se permettra donc de donner moins de détails dans la partie réalisation de la preuve.

Théorème 3.5.1

Soit $L \subseteq \mathbb{N} \times \mathbb{Z}$ un langage, alors L est l'ensemble des 1-périodes d'un SFT X si et seulement si $un(L) \in \mathbf{NSPACE}(n)$.

Remarque 3.5.1. On utilise ici le codage unaire suivant :

- quand $n \geq 0$, on code (m, n) par $1a^n b^{m-n}$ quand $m \geq n$ et par $1b^m a^{n-m}$ autrement.
- quand $n < 0$, on code (m, n) par $0a^n b^{m-n}$ quand $m \leq n$ et par $1b^m a^{n-m}$ autrement.

Lemme 3.5.2

Si X est un SFT 2-dimensionnel, alors $un(\mathfrak{P}_X^1) \in \mathbf{NSPACE}(n)$.

Preuve. Dans le lemme 3.1.2, on a vu comment les configurations d'un SFT de dimension 2 périodiques selon un vecteur v pouvaient être vues comme des SFTs de dimension 1. Les configurations périodiques selon v forment donc un SFT de dimension 1. On peut toujours représenter un SFT de dimension 1 comme un graphe, être 1-périodique est alors équivalent à l'existence de deux cycles mutuellement accessibles dans ce graphe. L'algorithme que l'on va donner va s'inspirer de ce fait

On donne maintenant pour un SFT X un algorithme permettant de décider si un vecteur v est une 1-période d'un point de X . On note r le rayon de X . L'algorithme est très similaire à celui du lemme 3.2.1, la différence est que cette fois il faut en plus vérifier l'existence d'une configuration dans laquelle v est l'unique vecteur de périodicité.

On va faire comme avant et deviner les lignes une à une, tout en se souvenant de suffisamment des dernières lignes pour pouvoir vérifier la validité au fur et à mesure. Il faut également que l'on vérifie qu'il existe deux manières de compléter différentes mais qui peuvent se raccorder. L'entrée de l'algorithme est $v = (m, n)$:

- On initialise un tableau P de taille $\max(m, n)$, tel que $P[i] = 1$ pour tout i et D à faux.
- On choisit non-déterministiquement deux tailles $t_1, t_2 \leq |\Sigma|^{2rp}$ (la longueur des "cycle" ou manières de compléter).

11	$x(3,11)$	$x(0,9)$	$x(1,9)$	$x(2,9)$	$x(3,9)$	$x(0,7)$	$x(1,7)$	$x(2,7)$	$x(3,7)$	$x(0,5)$	$x(1,5)$	$x(2,5)$	$x(3,5)$	$x(0,3)$	$x(1,3)$	$x(2,3)$	$x(3,3)$	$x(0,1)$	$x(1,1)$	$x(2,1)$	$x(3,1)$	$x(0,-1)$
10	$x(3,10)$	$x(0,8)$	$x(1,8)$	$x(2,8)$	$x(3,8)$	$x(0,6)$	$x(1,6)$	$x(2,6)$	$x(3,6)$	$x(0,4)$	$x(1,4)$	$x(2,4)$	$x(3,4)$	$x(0,2)$	$x(1,2)$	$x(2,2)$	$x(3,2)$	$x(0,0)$	$x(1,0)$	$x(2,0)$	$x(3,0)$	$x(0,-2)$
9	$x(3,9)$	$x(0,7)$	$x(1,7)$	$x(2,7)$	$x(3,7)$	$x(0,5)$	$x(1,5)$	$x(2,5)$	$x(3,5)$	$x(0,3)$	$x(1,3)$	$x(2,3)$	$x(3,3)$	$x(0,1)$	$x(1,1)$	$x(2,1)$	$x(3,1)$	$x(0,-1)$	$x(1,-1)$	$x(2,-1)$	$x(3,-1)$	$x(0,-3)$
8	$x(3,8)$	$x(0,6)$	$x(1,6)$	$x(2,6)$	$x(3,6)$	$x(0,4)$	$x(1,4)$	$x(2,4)$	$x(3,4)$	$x(0,2)$	$x(1,2)$	$x(2,2)$	$x(3,2)$	$x(0,0)$	$x(1,0)$	$x(2,0)$	$x(3,0)$	$x(0,-2)$	$x(1,-2)$	$x(2,-2)$	$x(3,-2)$	$x(0,-4)$
7	$x(3,7)$	$x(0,5)$	$x(1,5)$	$x(2,5)$	$x(3,5)$	$x(0,3)$	$x(1,3)$	$x(2,3)$	$x(3,3)$	$x(0,1)$	$x(1,1)$	$x(2,1)$	$x(3,1)$	$x(0,-1)$	$x(1,-1)$	$x(2,-1)$	$x(3,-1)$	$x(0,-3)$	$x(1,-3)$	$x(2,-3)$	$x(3,-3)$	$x(0,-5)$
6	$x(3,6)$	$x(0,4)$	$x(1,4)$	$x(2,4)$	$x(3,4)$	$x(0,2)$	$x(1,2)$	$x(2,2)$	$x(3,2)$	$x(0,0)$	$x(1,0)$	$x(2,0)$	$x(3,0)$	$x(0,-2)$	$x(1,-2)$	$x(2,-2)$	$x(3,-2)$	$x(0,-4)$	$x(1,-4)$	$x(2,-4)$	$x(3,-4)$	$x(0,-6)$
5	$x(3,5)$	$x(0,3)$	$x(1,3)$	$x(2,3)$	$x(3,3)$	$x(0,1)$	$x(1,1)$	$x(2,1)$	$x(3,1)$	$x(0,15)$	$x(1,15)$	$x(2,15)$	$x(3,15)$	$x(0,-1)$	$x(1,-1)$	$x(2,-1)$	$x(3,-1)$	$x(0,-3)$	$x(1,-3)$	$x(2,-3)$	$x(3,-3)$	$x(0,-7)$
4	$x(3,4)$	$x(0,2)$	$x(1,2)$	$x(2,2)$	$x(3,2)$	$x(0,0)$	$x(1,0)$	$x(2,0)$	$x(3,0)$	$x(0,14)$	$x(1,14)$	$x(2,14)$	$x(3,14)$	$x(0,-2)$	$x(1,-2)$	$x(2,-2)$	$x(3,-2)$	$x(0,-4)$	$x(1,-4)$	$x(2,-4)$	$x(3,-4)$	$x(0,-8)$
3	$x(3,3)$	$x(0,1)$	$x(1,1)$	$x(2,1)$	$x(3,1)$	$x(0,15)$	$x(1,15)$	$x(2,15)$	$x(3,15)$	$x(0,13)$	$x(1,13)$	$x(2,13)$	$x(3,13)$	$x(0,-3)$	$x(1,-3)$	$x(2,-3)$	$x(3,-3)$	$x(0,-5)$	$x(1,-5)$	$x(2,-5)$	$x(3,-5)$	$x(0,-9)$
2	$x(3,2)$	$x(0,0)$	$x(1,0)$	$x(2,0)$	$x(3,0)$	$x(0,14)$	$x(1,14)$	$x(2,14)$	$x(3,14)$	$x(0,12)$	$x(1,12)$	$x(2,12)$	$x(3,12)$	$x(0,-4)$	$x(1,-4)$	$x(2,-4)$	$x(3,-4)$	$x(0,-6)$	$x(1,-6)$	$x(2,-6)$	$x(3,-6)$	$x(0,-10)$
1	$x(3,1)$	$x(0,15)$	$x(1,15)$	$x(2,15)$	$x(3,15)$	$x(0,13)$	$x(1,13)$	$x(2,13)$	$x(3,13)$	$x(0,11)$	$x(1,11)$	$x(2,11)$	$x(3,11)$	$x(0,-5)$	$x(1,-5)$	$x(2,-5)$	$x(3,-5)$	$x(0,-7)$	$x(1,-7)$	$x(2,-7)$	$x(3,-7)$	$x(0,-11)$

FIGURE 3.15 – En gris la dernière ligne devinée, et en hachuré la zone que l'on doit conserver en mémoire. Ici le vecteur de périodicité est $(m, n) = (4, 2)$ et le rayon du SFT $r = 1$.

- On choisit non-déterministiquement $2r$ lignes horizontales $(l_i)_{0 \leq i \leq r-1}$ de longueur m , c'est à dire $2r \times m$ symboles de Σ .
- Pour tout $2r < i \leq t_1$, on choisit non déterministiquement une ligne l_i de longueur m (donc m symboles) et on vérifie qu'aucun motif de support B_r dans les lignes l_i, \dots, l_{i-2r-n} ne contienne un motif interdit. Il suffit de garder à chaque étape les $2r$ dernière lignes et $2r$ symboles des n lignes précédentes, on pourra se référer à la figure 3.15 afin d'avoir une figure explicative.
- On effectue une deuxième fois les deux étapes précédentes, en parallèle. Pour la deuxième, on choisit une ligne l'_i pour $2r < i \leq t_2$ cette fois-ci. Si à un moment, pour un i , les lignes devinées l_i et l'_i sont différentes, alors on met D à vrai.
- A chaque étape précédente, on vérifie s'il y a un vecteur de périodicité (m', n') avec $k(m', n') = (m, n)$, si ce n'est pas le cas, alors $P[k] \leftarrow 0$.
- Une fois les dernières lignes l_{t_1}, l'_{t_2} devinées, on vérifie que les successions de lignes suivantes ne contiennent aucun motif interdit :
 - $l_{t_1-2r-n}, \dots, l_{t_1}, l_0, \dots, l_{2r}$
 - $l'_{t_2-2r-n}, \dots, l'_{t_2}, l'_0, \dots, l'_{2r}$
 - $l_{t_1-2r-n} \dots l_{t_1} l'_0 \dots l'_{2r}$
 - $l'_{t_2-2r-n} \dots l'_{t_2} l_0 \dots l_{2r}$
 On vérifie ainsi l'existence de deux cycles mutuellement accessibles.
- On rejette s'il existe $k, m', n' < m$ tels que $k(m', n') = (m, n)$ et $P[k] = 1$ ou si D est à faux.

Cet algorithme nécessite de conserver $4rm + 2rn$ symboles ainsi que le tableau P à chaque étape. \square

Lemme 3.5.3

Pour toute constantes c , il existe un SFT 2-dimensionnel Y_c tel que les points 1-périodiques de vecteur (m, n) soient formés de rectangles $m \times c^{m-1}$ à bords marqués, comme en figure 3.16. De plus Y_c admet comme 1-périodes tout (m, n) tel que $0 < n < m$.

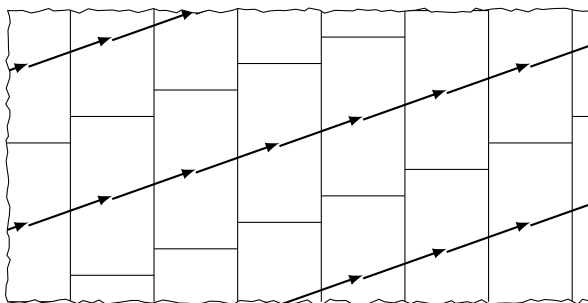


FIGURE 3.16 – Les rectangles marqués des configurations périodiques.

Preuve. Comme dans les constructions précédentes, la construction va se faire par étapes de superposition. On aura donc à nouveau diverses composantes A, C_c, R, S :

- La première composante A est basée sur un SFT apériodique Est-déterministe W , son alphabet est $\Sigma_A = (\Sigma_W \times \{\cdot, \blacksquare\}) \cup \{\blacksquare, \blacksquare, \blacksquare, \blacksquare\}$, on appelle les symboles de Σ_W des *blancs*. Les symboles que l'on a ajouté permettent à nouveau de briser la périodicité, les règles pour A sont les suivantes :
 - Les symboles blancs peuvent avoir un symbole \blacksquare au dessus, ou rien.
 - Les règles entre les symboles de Σ_W sont inchangées.
 - Les règles entre les symboles $\{\blacksquare, \blacksquare, \blacksquare, \blacksquare, \blacksquare\}$ sont des règles de Wang.
 - Seules les faces blanches des symboles $\{\blacksquare, \blacksquare, \blacksquare, \blacksquare, \blacksquare\}$ peuvent être mises à côté de symboles blancs.

A ce stade, les configurations périodiques ont nécessairement soit une infinité de lignes verticales ou de lignes horizontales. Les lignes verticales peuvent éventuellement être jointes par des lignes horizontales finies.

- La composante C_c est un compteur qui fonctionne exactement de la même manière que dans la preuve du lemme 3.2.2. Une fois que l'on a ajouté C_c à A les points ayant un vecteur de périodicité contiennent nécessairement une infinité de lignes verticales. Deux lignes verticales espacées de k sont alors jointes par des lignes horizontales à distance 2^{k-1} .
- La composante R va permettre de s'assurer que la distance entre les différentes lignes verticales dans les configurations périodiques est constante. Pour faire cela on projette les lignes horizontales à gauche et à droite, et entre deux projections on remet un compteur C_c . On se sert également de ces projections afin de faire en sorte que le décalage entre les rectangles d'une colonne de

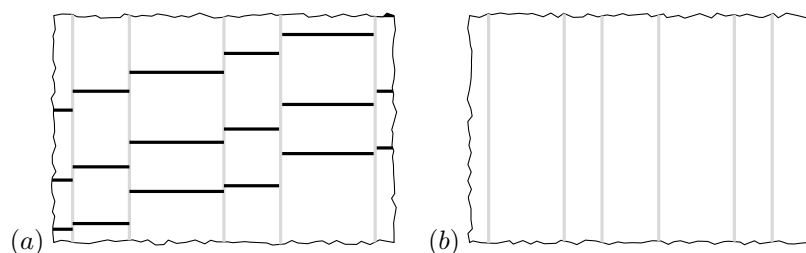


FIGURE 3.17 – Il y a plusieurs possibilités pour les points périodiques de A : Une infinité de lignes verticales (b) éventuellement jointes par des lignes horizontales (a).

symboles blancs à l'autre soit constant. La figure 3.18 montre comment la composante R agit.

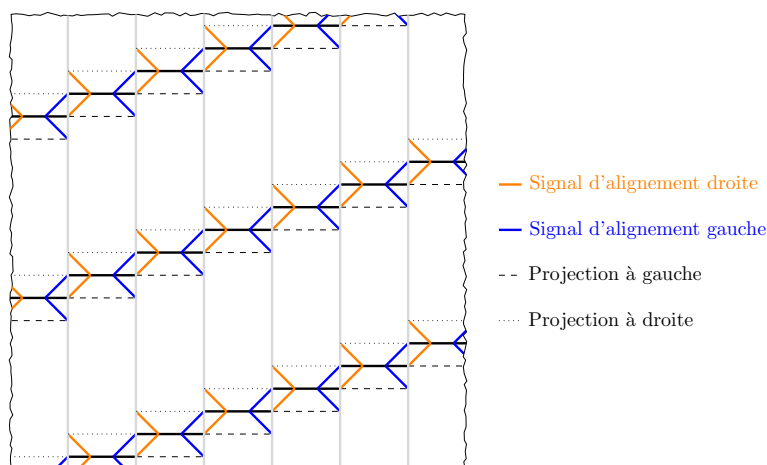


FIGURE 3.18 – Comment fonctionne la composante R . A chaque ligne horizontale, une projection est envoyée jusqu'à la ligne verticale la plus proche à gauche et à droite. Entre deux projections à gauche (resp. à droite) il y a un compteur C_c . On envoie également un signal à chaque extrémité d'une ligne horizontale, afin que les rectangles soient toujours décalés de la même hauteur.

À ce stade, les points périodiques sont nécessairement constitués de rectangles de taille identique "translatés" par un vecteur (m', n') d'une colonne à l'autre. (m', n') n'est pas nécessairement une période, mais il existe k tel que $k(m', n')$ soit un vecteur de périodicité.

- On fait maintenant en sorte que la largeur des rectangles m et leur décalage n soit le vecteur un vecteur de périodicité. On fait cela en transmettant la première ligne verticale infinie de symboles blancs de chaque colonne de rectangles vers la droite en la décalant de n vers le haut. La dernière composante S fait exactement cela. Il y a deux parties à S , une première W' qui est une copie des symboles de W et une autre qui permet de savoir quels symboles voisins doivent être égaux, que l'on peut voir en figure 3.19. La figure suffit à en inférer les règles. Comme précédemment, les symboles sur W et sur W'

juste à droite d'une ligne verticale sont identiques.

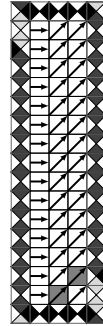


FIGURE 3.19 – Les symboles \blacksquare sont superposés uniquement aux signaux permettant de synchroniser les décalages. Les règles pour la copie de symboles sur W' sont déterminées par les flèches : un symbole sur W' est égal au symbole vers lequel il pointe.

- La dernière étape consiste à faire en sorte qu'il existe des points 1-périodiques pour tous les vecteurs de périodicité (m, n) avec $0 < n < m$. Pour cela, il suffit d'ajouter deux couleurs possibles aux rectangles, bleu et jaune : celle couleur est transmise aux translatés par (m, n) de chaque rectangle.

Vérifions maintenant que les points 1-périodiques ont bien la bonne forme. Soit x un point 1-périodique de 1-période (m, n) où $0 < n < m$. Alors, x a nécessairement une infinité de lignes verticales grâce à A . Deux lignes verticales se faisant face sont jointes par des lignes horizontales grâce à C_c , de plus R force les lignes verticales à être espacées régulièrement et à être décalées régulièrement aussi. Et T force l'arrière plan de chaque colonne de symboles blancs à être identique mais décalé comme les rectangles. Donc x est périodique selon (m', n') , où m' est la largeur des rectangles et n' le décalage vertical entre les rectangles d'une colonne à sa voisine de droite. Mais (m', n') est le plus petit vecteur de périodicité, donc $(m', n') = (m, n)$.

Inversement, étant donné (m, n) il est facile de construire un point 1-périodique : une configuration bien formée avec des rectangles $m \times 2^{m-1}$ et on met une ligne de direction (m, n) de rectangles en bleu et toutes les autres en jaune. \square

Lemme 3.5.4

Soit $L \in \mathbb{N} \times \mathbb{Z}$ un langage tel que $un(L) \in \mathbf{NSPACE}(n)$, alors il existe un SFT X tel que $L = \mathfrak{P}_X^1$.

Preuve. Étant donné une machine M qui reconnaît le langage L en temps c^n , pour une certaine constante c , n étant l'entrée. On peut facilement faire un SFT qui reconnaît le sous langage de L formé des (m, n) tels que $0 < n < m$ en codant la machine M dans le SFT Y_c , tout en synchronisant bien les transitions non-déterministes. On peut faire une union disjointe de SFTs basés sur des constructions quasiment identiques afin d'obtenir les autres sous-cas $0 < m < n$, $m = 0$, $n = 0$, $n = m$ et $n < 0 < m$ (il y a alors deux sous-cas $|n| < |m|$ et $|m| < |n|$). \square

Il y a deux manières d'envisager la généralisation en dimensions supérieures du théorème 3.5.1 : la 1-périodicité en dimension 2 est en effet à la fois la $d-1$ -périodicité et à la 1-périodicité en dimension d . Caractériser l'ensemble \mathfrak{P}_X^1 d'un SFT de dimension $d > 2$ sera indécidable de manière triviale. Si l'on s'intéresse à la $d-1$ -périodicité en dimension d , on a en revanche une notion de périodicité qui se rapporte à nouveau à l'étude d'un SFT de dimension 1 et qui pourrait être caractérisée à l'aide de classes de complexité.

3.6 Ensembles de Périodes des sous-shifts sofiques et effectifs

On s'intéresse maintenant au cas des sous-shifts sofiques et effectifs, où l'on va parvenir à caractériser les d -uplets $(n_1, \dots, n_d) \in \mathbb{N}^{*d}$ tels qu'il y ait un point x avec $\Gamma_x = n_1\mathbb{Z} \times \dots \times n_d\mathbb{Z}$. Cette notion est plus générale que la périodicité forte, étudiée dans le cas des SFTs. On notera \mathfrak{P}_X^r l'ensemble des tels d -uplets du sous-shift X . En dimension 2, cette notion de périodicité correspond à un treillis "rectangulaire", tandis que la périodicité forte correspond à un treillis "carré". On obtient le théorème suivant :

Théorème 3.6.1

Soit $L \subseteq \mathbb{N}^{*d}$, il existe un sous-shift effectif tel que $\mathfrak{P}_X^r = L$ si et seulement si L est Π_1^0 , c'est à dire co-récurivement énumérable.

Lemme 3.6.2

Soit X un sous-shift effectif de dimension d , \mathfrak{P}_X^r est Π_1^0 .

Preuve. Soit X un sous-shift effectif, et M la machine de Turing énumérant ses motifs interdits. On construit une machine M' prenant en entrée un d -uplet et s'arrêtant uniquement s'il n'existe pas de point x dans X tel que $\Gamma_x = n_1\mathbb{Z} \times \dots \times n_d\mathbb{Z}$. M' commence par deviner en parallèle toutes les complétions possibles de l'hyperrectangle $R = \llbracket 0, n_1 - 1 \rrbracket \times \dots \times \llbracket 0, n_d - 1 \rrbracket$. Puis elle simule M et au fur et à mesure que les motifs interdits sont énumérés, elle vérifie que le motif énuméré n'apparaît nulle part dans aucune complétion. On étend bien entendu chaque complétion périodiquement, de manière à pouvoir vérifier que le motif énuméré n'apparaît en aucune position de l'hyperrectangle R . Quand on a trouvé pour chaque complétion un motif interdit qui y apparaît, on s'arrête, sinon la machine continue à énumérer des motifs interdits sans jamais s'arrêter. \square

Lemme 3.6.3

Soit $L \subseteq \mathbb{N}^{*d}$, si L est Π_1^0 , alors il existe un sous-shift sofique X de dimension d tel que $\mathfrak{P}_X^r = L$.

Preuve. On va détailler la preuve en dimension 2, la preuve en dimensions supérieures étant quasiment identique. On a un langage $L \subseteq \mathbb{N}^2$, co-récursivement énumérable et on veut construire un sous-shift sofique X tel que $\mathfrak{P}_X^r = L$. On détaille maintenant la construction d'un sous-shift sofique réalisant le sous-ensemble $\{(n_1, n_2) \mid (n_1, n_2) \in L \text{ et } n_1 > n_2\}$ de L . La construction va à nouveau utiliser une base apériodique que l'on rendra périodique, de la forme souhaitée, en ajoutant par dessus des couches qui ne seront pas périodiques. Il suffira alors d'effectuer une projection pour obtenir le résultat.

- La première composante A consiste en un SFT apériodique NO-déterministe W auquel on ajoute trois symboles $\underline{\quad}, \oplus, \mathbf{I}$. Les règles sont quasiment les mêmes que pour A' de la composante A du lemme 3.3.4 : Au dessus d'un \mathbf{I} ou d'un \oplus ne peuvent apparaître qu'un \mathbf{I} ou un \oplus . À gauche et à droite d'un \oplus ou d'un $\underline{\quad}$ ne peuvent apparaître qu'un \oplus ou un $\underline{\quad}$. Pour que cette couche soit périodique, il faut qu'une infinité de lignes de \mathbf{I} et/ou une infinité de lignes de $\underline{\quad}$ apparaisse. Si il y a une infinité de lignes de \mathbf{I} et une infinité de lignes de $\underline{\quad}$, leurs croisements sont faits de \oplus et elles forment des rectangles dont les tailles ne sont pas nécessairement identiques.
- La deuxième composante P va nous permettre de forcer les dimensions de tous ces rectangles soient les mêmes et à ce que ce soient des dimensions apparaissant dans L . Pour cela, on veut des lignes identiques formées de mots $b^m a^{n-m}$, avec $n > m$:

$$\dots b^m a^{n-m} b^m a^{n-m} b^m a^{n-m} \dots$$

On peut voir ces lignes comme un sous-shift de dimension 1. Bien entendu, par compacité, on ne pourra pas éviter le cas où la ligne est formée uniquement de b ou uniquement de a . Ce sous-shift est effectif, donc en utilisant le théorème 1.1.4, on sait que l'on peut réaliser un SFT qui a une composante qui est son relèvement.

On donne maintenant les règles de superposition : un b apparaissant après un a doit nécessairement être superposé soit à un \mathbf{I} , soit à un \oplus .

À ce stade, la première composante est périodique dans deux cas : soit la composante P est uniforme et ne contient que des a ou b et alors elle doit avoir une infinité de lignes de $\underline{\quad}$. Soit elle a une infinité de lignes de \mathbf{I} distantes de n , avec n tel qu'il existe $(n, m) \in L$, avec $n > m$.

- La composante R va forcer l'apparition de lignes horizontales de $\underline{\quad}$ quand il y a une infinité de lignes verticales de \mathbf{I} . R sera donc constitué des tuiles de Wang suivantes : $\{\begin{smallmatrix} \color{red}\blacksquare, \color{blue}\blacksquare, \color{red}\blacksquare, \color{blue}\blacksquare, \color{red}\blacksquare, \color{blue}\blacksquare, \color{red}\blacksquare, \color{blue}\blacksquare, \color{red}\blacksquare, \color{blue}\blacksquare \end{smallmatrix}\}$. Les superpositions seront les suivantes :
 - On ne peut superposer $\color{blue}\blacksquare$ que sur \mathbf{I} .
 - On ne peut superposer $\color{red}\blacksquare$ que sur \oplus .
 - On ne peut superposer $\color{red}\blacksquare$ et $\color{blue}\blacksquare$ que sur un a à droite d'un b .
 - On ne peut superposer $\color{red}\blacksquare$ que sur $\underline{\quad}$.
 - $\color{yellow}\blacksquare, \color{red}\blacksquare, \color{blue}\blacksquare$ ne peuvent être superposés que sur b .
 - $\color{blue}\blacksquare$ ne peut être superposé que sur a .

On pourra se référer à la figure 3.20 pour mieux comprendre comment cette composante se superpose aux autres et forme des rectangles.

Les cas possibles quand la composante A est périodique sont les suivants : soit on a des rectangles uniformes de dimensions $n \times m$, avec $n > m$ et $(n, m) \in L$, soit il y a une infinité de lignes horizontales de $\underline{\quad}$. Dans le cas des rectangles, $(n, 0), (0, m)$ ne génère pas nécessairement Γ_x , l'arrière plan

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

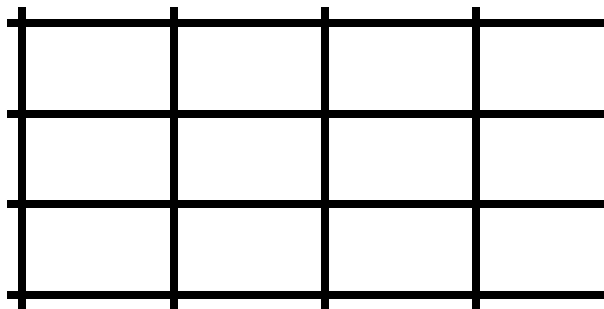
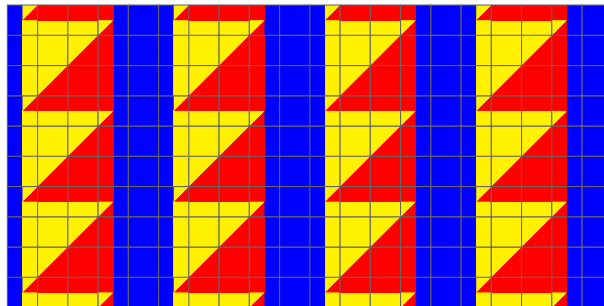


FIGURE 3.20 – Comment les couches A, P et R sont superposées : tout en haut la composante P , juste en dessous R et encore en dessous A . Les lignes verticales de \blacksquare sont superposées sur les premiers symboles b apparaissant après a . La composante R permet d'ajuster la hauteur des rectangles par rapport au nombre de b .

apériodique à l'intérieur de chaque rectangle n'étant pas nécessairement le même.

- La composante V va permet de forcer la première colonne à l'intérieur de chaque rectangle sur la composante A à avoir les mêmes symboles. Le sous-shift de dimension 1 formé à partir de l'alphabet $\Sigma_W \cup \{\#\}$ dont les points sont ceux de la forme :

$$\cdots \#w_1 \dots w_k \#w_1 \dots w_k \#w_1 \dots w_k \# \cdots$$

c'est à dire les points périodiques formés d'un mot sur l'alphabet de W répété périodiquement dont chaque occurrence est séparée par un $\#$. On relève ce sous-shift en dimension 2 verticalement. Et on force les $\#$ à être superposés uniquement à des \blackplus ou des \blackminus . On force aussi les symboles sur A à droite d'un \mathbf{I} à être identiques à ceux sur V .

- La dernière composante H est formée un peu de la même manière que V , on considère le même sous-shift mais on ajoute un second symbole en plus :

$$\cdots \#(w_1, t_1) \dots (w_k, t_k) \#(w_1, t_1) \dots (w_k, t_k) \#(w_1, t_1) \dots (w_k, t_k) \# \cdots$$

Les w_i sont des symboles de Σ_W et les t_i des symboles d'un sous-shift effectif de dimension 1 apériodique. On conserve les règles de ce sous-shift apériodique entre les symboles t_i qui ne sont pas séparés par un $\#$, c'est à dire que si aucun $\#$ n'apparaît, alors le point n'est pas périodique. On peut maintenant relever ce sous-shift en dimension 2 horizontalement.

On force alors les symboles $\#$ à être superposés à des \blackplus ou \mathbf{I} uniquement. Et on force les symboles de la composante A apparaissant juste en dessous d'un \blackminus à être identiques à ceux sur H .

Maintenant, si l'on considère les points périodiques possibles sur la composante A :

- Il y a les points x formés de rectangles de taille $n \times m$, avec $n > m$ et $(n, m) \in L$, ils sont périodiques et $\Gamma_x = n\mathbb{Z} \times m\mathbb{Z}$.
- Il y a les points formés d'une infinité de lignes horizontales de \blackminus sans ligne verticale de \mathbf{I} .

On obtient notre sous-shift sofique en projetant la première composante A et la dernière composante H : cette dernière élimine la classe de points périodiques formés de lignes horizontales de \blackminus uniquement car il est impossible que ceux-ci soient périodiques horizontalement à cause de la deuxième couche de symboles de H qui est rend le sous-shift apériodique horizontalement. \square

On peut noter que la preuve de ce résultat en dimension 1 est relativement simple : il suffit de prendre un sous-shift effectif apériodique que l'on casse avec des $\#$ qui ne peuvent être espacés qu'uniformément d'une distance $n \in L$.

3.7 Remarques sur les périodes en tant que langages

On peut également considérer les ensembles de périodes comme des langages. Un langage de période est un langage qui peut être reconnu par un sous-shift : un entier est accepté s'il est période du sous-shift. En tant que langages, la question de la stabilité par les opérations usuelles d'union, intersection, complémentation est importante. Nos caractérisation nous permettent d'apporter

des réponses dans une certaine mesure. La stabilité par union est très facile à prouver pour toutes les notions de périodicité : pour deux langages L_1, L_2 , il suffit de prendre l'union disjointe des deux sous-shifts X_1, X_2 dont ce sont les ensembles de périodes pour obtenir $L_1 \cup L_2$.

On a le résultat suivant sur la stabilité des langages de périodes horizontales de SFTs de dimension 2 :

Corollaire 3.7.1*Langages de périodes horizontales*

Les langages de périodes horizontales de SFTs sont stables par union, intersection et complémentation.

Preuve. La stabilité par union et intersection de la classe de complexité $\mathbf{NSPACE}(n)$ est triviale. D'après Immerman [Imm88], Szelepcsényi [Sze88], on sait aussi que $\mathbf{NSPACE}(n)$ est close par complémentation. \square

Pour la stabilité par intersection, une construction par produit cartésien, comme on aurait envie de le faire, ne fonctionnera pas : si l'on prend par exemple $\mathfrak{P}_X^h = \{2\}$ et $\mathfrak{P}_Y^h = \{3\}$, le produit cartésien $X \times Y$ n'aura pas comme ensemble de périodes \emptyset mais $\mathfrak{P}_{X \times Y}^h = \{6\}$. Ce résultat est également valable pour les langages de 1-périodes de SFTs de dimension 2.

Dans le cas des périodes fortes, le résultat est moins fort :

Corollaire 3.7.2*Langages de périodes fortes*

Les langages de périodes fortes de SFTs sont stables par union, intersection. Si $\mathbf{NE} = \mathbf{coNE}$, alors ils sont stables par complémentation.

Ce théorème est à nouveau une conséquence directe de la caractérisation que nous avons prouvée sur les périodes fortes. Il est important de noter que si l'on a deux SFTs X, Y de dimension d , le langage $\mathfrak{P}_X \cap \mathfrak{P}_Y$ est un langage de période forte, mais le SFT le reconnaissant n'est pas forcément de dimension d .

Chapitre 4

Difficulté de la factorisation et de la conjugaison

Sommaire

4.1	Conjugaison et égalité	84
4.1.1	Sous-shifts de type fini	84
4.1.2	Sous-shifts effectifs et sofiques	87
4.2	Factorisation	89
4.2.1	La factorisation est Σ_3^0	90
4.2.2	La factorisation est Σ_3^0 -dure	90

Alors que jusqu'à présent nous avons étudié des invariants de conjugaison, nous nous intéressons ici à la complexité de la conjugaison elle-même. Dans un second temps nous nous intéresserons aussi à la complexité de la factorisation, qui est un problème important d'un point de vue dynamique.

En dimension 1, le problème de décision associé à la conjugaison est peut-être le problème ouvert le plus connu de la dynamique symbolique : on ne sait toujours pas à ce jour s'il est décidable ou pas, voir la liste de problèmes ouverts de Boyle [Boy08]. Si l'on se place sur \mathbb{N} au lieu de \mathbb{Z} , la réponse est connue : la conjugaison est décidable, voir Williams [Wil73]. Aubrun et Béal [AB09] ont prouvé le même résultat sur d'autres monoïdes.

En dimension 2, le problème est indécidable, cela découle du fait que savoir si un SFT est vide est indécidable. On prouve ici un résultat un peu plus fort, en donnant sa complexité exacte dans la hiérarchie arithmétique :

Théorème 4.0.3

En dimension $D \geq 2$, pour tout SFT X fixé, étant donné un SFT Y en entrée, décider si Y et X sont conjugués est Σ_1^0 -complet.

Ce résultat est intéressant dans le sens où décider si deux SFTs sont conjugués n'est pas plus dur que décider si un SFT est vide, ou de manière équivalente si son langage est vide. Une question ouverte intéressante est de savoir si décider la conjugaison en ayant le langage du SFT en oracle reste indécidable.

Nous avons également trouvé la complexité exacte de la conjugaison dans le cas des sous-shifts effectifs de dimension $d \geq 1$ et sofiques de dimension $d \geq 2$. Celle-ci est plus élevée :

Théorème 4.0.4

Étant donnés deux sous-shifts effectifs (resp. sofiques de dimension $d \geq 2$). X et Y en entrée, décider si X et Y sont conjugués est Σ_3^0 -complet.

Le second problème que l'on étudie ici, la factorisation, est partiellement ouvert en dimension 1. Quand les deux SFTs X, Y donnés en entrée n'ont pas la même entropie, le problème a été résolu par Boyle [Boy83] : supposons que $h(X) > h(Y)$, alors X se factorise sur Y si et seulement si pour toute période n de X , il existe un point de période d divisant n dans Y . En dimensions supérieures, le problème est bien entendu indécidable, pour la même raison que la conjugaison. On prouve ici cependant que sa complexité est bien plus élevée que celle de la conjugaison dans le cas des SFTs et la même dans le cas des sofiques/effectifs :

Théorème 4.0.5

Étant donnés deux SFTs de dimension $d \geq 2$ (resp. sous-shifts sofiques/effectifs) X, Y en entrée, décider si X se factorise sur Y est Σ_3^0 -complet.

Il est assez surprenant de constater que le saut en complexité entre les SFT et les sofiques/effectifs qui a lieu pour la conjugaison n'existe plus pour la factorisation.

On peut également se demander quelle est la complexité exacte du plongement, qui a aussi une grande importance en dynamique symbolique.

Nous commençons, en section 4.1, par prouver le théorème sur la conjugaison, puis nous nous intéressons à la factorisation en section 4.2. Une des preuves de ce chapitre nécessite d'être familier avec la construction formant une grille irrégulière de la section 2.2.1 du chapitre 2.

4.1 Conjugaison et égalité

4.1.1 Sous-shifts de type fini

Il est important de commencer cette section par un rappel sur la différence entre les notions de motif extensible et admissible (définis en section 1.1.1) :

Rappel 4.1.1. Soit \mathcal{F} une famille finie de motifs interdits et $X = \mathcal{X}_{\mathcal{F}}$ le sous-shift associé : un motif est admissible quand il ne contient pas de motif interdit par \mathcal{F} et un motif est extensible quand il apparaît dans un point de X .

Si la famille \mathcal{F} est finie cela signifie, au niveau décidabilité, que décider si un motif est extensible est Π_1^0 , tandis que décider si un motif est admissible est calculable.

Avant de nous intéresser à la conjugaison, nous nous intéressons à l'égalité de sous-shifts, en commençant par prouver un lemme légèrement plus généraliste que nécessaire : la complexité de l'égalité en sera une conséquence directe et nous pourrions réutiliserons le lemme dans la suite.

Lemme 4.1.1

Étant donnés deux SFTs X, Y et une fonction F continue commutant avec le shift, savoir si $F(X) \subseteq Y$ est Σ_1^0 .

Remarque 4.1.1. Les SFTs sont donnés en entrée sous la forme d'un couple (Σ, \mathcal{F}) : l'alphabet et la famille de motifs interdits. Une fonction locale $F : \Sigma_X^{\mathbb{B}^r} \rightarrow \Sigma_Y$ est codée par une liste L d'éléments $(p \rightarrow i)$, où i est une lettre de Σ_Y et est l'image du motif $p \in \Sigma_X^{\mathbb{B}^r}$.

Preuve. Il est clair que $F(X) \subseteq Y$ si et seulement si $F(X)$ ne contient aucun point contenant un motif interdit de Y . On montre que ceci est équivalent à la proposition Σ_1^0 suivante :

Il existe un rayon $r \geq \max(r_F + r_Y, r_X)$ tel que tout r -bloc M admissible pour X , $F(M)$ ne contient pas de motif interdit de Y en $\mathbf{0}$.

On montre ce résultat par contraposition dans les deux directions :

- \Rightarrow Soit $x \in X$ une configuration telle que $F(x)$ contient un motif interdit. Alors pour tout rayon $r \geq \max(r_F + r_Y, r_X)$, il existe un r -bloc M extensible et donc admissible tel que $F(M)$ contient un motif interdit de Y en $\mathbf{0}$.
- \Leftarrow Inversement, si pour tout rayon $r \geq \max(r_F + r_Y, r_X)$, il existe un r -bloc M admissible pour X tel que $F(M)$ ait un motif interdit de Y en $\mathbf{0}$, alors par compacité il existe un point $x \in X$ tel que $F(x)$ ait un motif interdit de Y en $\mathbf{0}$.

□

Corollaire 4.1.2

Étant donnés deux SFTs X, Y , savoir si $X = Y$ est Σ_1^0 .

La réduction qui suit nous permet de montrer que la conjugaison et l'égalité de sous-shift sont toutes les deux Σ_1^0 -dures.

Lemme 4.1.3

Pour tout SFT X fixé, étant donné Y en entrée, savoir si X et Y sont conjugués est Σ_1^0 -dur.

Corollaire 4.1.4

Pour tout SFT X , étant donné Y , savoir si X et Y sont égaux est Σ_1^0 -dur.

Preuve du lemme 4.1.3. On réduit le problème à $\mathbf{0}'$, le problème de la halte. Étant donné une machine de Turing M , on construit un SFT Y_M tel que Y_M est conjugué à X si et seulement si M s'arrête.

Prenons R_M le SFT de Robinson dans lequel sont codés les calculs de M : R_M est vide si et seulement si M s'arrête. Soit F le full-shift sur un alphabet avec strictement plus de symboles que celui de X . On prend $Y_M = X \sqcup (R_M \times F)$.

Si M s'arrête, alors $Y_M = X$ et est donc conjugué à X . Inversement, si M ne s'arrête pas, alors

$$h(Y_M) \geq \log \#\Sigma_F > h(X),$$

comme l'entropie est invariante par conjugaison, Y_M n'est pas conjugué à X . \square

On suffit donc maintenant prouver que la conjugaison est dans Σ_1^0 afin de terminer la preuve du théorème 4.0.3.

Lemme 4.1.5

Étant donnés deux SFTs X, Y en entrée, décider si X et Y sont conjugués est Σ_1^0 .

Preuve. Pour décider si deux SFTs X et Y sont conjugués, il faut vérifier s'il existe deux fonctions locales $F : \Sigma_X^{B_{r_F}} \rightarrow \Sigma_Y$ et $G : \Sigma_Y^{B_{r_G}} \rightarrow \Sigma_X$ telles que les fonctions globales vérifient $F|_X \circ G|_Y = \text{id}|_X$ et $G|_Y \circ F|_X = \text{id}|_Y$. Ces fonctions étant locales, les deviner correspond à un quantificateur existentiel du premier ordre.

On prouve que X et Y sont conjugués si et seulement si la proposition Σ_1^0 suivante est vraie :

Il existe F, G et $k > (\max(r_X + r_Y) + r_F + r_G)$ tel que $F(X) \subseteq Y$, $G(Y) \subseteq X$ et :

- pour tout k -bloc b , si b est admissible pour X , alors $G \circ F(b)_0 = b_0$
- pour tout k -bloc b , si b est admissible pour Y , alors $F \circ G(b)_0 = b_0$

On ne prouve que le cas $G \circ F$, l'autre cas étant identique. La preuve se fait par contraposition dans les deux sens :

- Soit $x \in X$ un point tel que $G \circ F(x) \neq x$, on peut supposer que la différence a lieu en $\mathbf{0}$ en translatant. Pour tout k , il existe un motif extensible (donc admissible) b de taille k tel que $(G \circ F(b))_0 \neq b_0$.
- Inversement, s'il existe une suite b_k de k -blocs admissibles tels que $(G \circ F(b_k))_0 \neq (b_k)_0$, alors par compacité on peut en extraire une suite convergent vers un point $x \in X$ qui, par construction, diffère de son image en $\mathbf{0}$.

Comme on a vu au lemme 4.1.1 que vérifier si $F(X) \subseteq Y$ était Σ_1^0 , on a le résultat. \square

Les lemmes 4.1.3 et 4.1.5 prouvent le théorème 4.0.3.

4.1.2 Sous-shifts effectifs et sofiques

Pour les sous-shifts effectifs, la complexité du problème est plus élevée : il n'y a plus de différence de complexité entre vérifier si un motif est admissible ou extensible : dans les deux cas le problème est Π_1^0 . Cela entraîne une augmentation de la complexité pour l'analogie du lemme 4.1.1

Lemme 4.1.6

Étant donnés deux sous-shifts effectifs X, Y et une fonction F continue commutant avec le shift, savoir si $F(X) \subseteq Y$ est Π_2^0 .

Preuve. $F(X) \subseteq Y$ si l'image de tout motif extensible de X est un motif extensible de Y et peut donc être exprimée par la proposition :

Pour tout motif M de rayon r , M est extensible pour $X \Rightarrow F(M)$ est extensible pour Y .

Qui est clairement dans Π_2^0 . □

Corollaire 4.1.7

Étant donnés deux sous-shifts effectifs X, Y et une fonction F continue commutant avec le shift, savoir si $X = Y$ est Π_2^0 .

Il ne nous reste plus qu'à montrer que le problème est Π_2^0 -dur afin d'avoir la complétude.

Théorème 4.1.8

Étant donnés deux sous-shifts effectifs X, Y , décider si $X = Y$ est Π_2^0 -dur.

Preuve. Pour montrer que le problème est Π_2^0 -dur, on part du problème TOTAL qui est Π_2^0 -complet. On prend les deux sous-shifts effectifs de dimension 1 suivants :

- Soit M une machine de Turing, X_M est le sous-shift sur les symboles $\{\#, 0\}$ où l'on interdit tous les $\#0^n\#$ tels que $M(n) \downarrow$.
- Y est le sous-shift sur l'alphabet $\{\#, 0\}$ où l'on interdit tous les mots $\#0^n\#$, pour $n \in \mathbb{N}$, de sorte que le sous-shift soit composé des points :

$\dots 00000000 \dots$ et $\dots 00000\#000000 \dots$

Le sous-shift X est égal au sous-shift Y si et seulement si la machine M s'arrête sur toutes les entrées : si la machine M ne s'arrête pas sur n , alors le sous-shift X_M contient le point périodique $\dots \#0^n\#0^n\# \dots$ qui n'est pas dans Y . □

En utilisant le théorème 1.1.4 permettant de relever un sous-shift effectif de dimension 1 en sous-shift sofique de dimension 2, on obtient le corollaire suivant :

Corollaire 4.1.9

Étant donnés deux sous-shifts sofiques X, Y de dimension 2, savoir si $X = Y$ est Π_2^0 -complet.

Retournons maintenant au problème de la conjugaison : une adaptation facile de la preuve du lemme 4.1.5 à partir du lemme 4.1.6 nous permet de montrer que la conjugaison est dans Σ_3^0 . Il ne reste donc qu'à prouver la Σ_3^0 -dureté. Nous faisons cela en réduisant à partir de COFIN, l'ensemble des machines de Turing qui ne s'arrêtent pas sur un nombre fini d'entrées, voir la section 1.2.2 du chapitre 1.

Lemme 4.1.10

Étant donnés deux sous-shifts effectifs (resp. sofiques de dimension $d \geq 2$) X et Y en entrée, décider si X et Y sont conjugués est Σ_3^0 -dur.

Preuve. Nous donnons une réduction pour les sous-shifts effectifs de dimension 1, le théorème de relèvement des sous-shifts effectifs en sofiques permet d'avoir la réduction pour les sofiques de dimension $d \geq 2$. Étant donné une machine de Turing M , on construit deux sous-shifts X_M et Y_M sur l'alphabet $\{\#, 0, 1\}$:

- X_M : on interdit les mots $\#1, 1\#, 10, 0\#$, les mots $\#0^k 1$ quand k n'est pas de la forme 2^{i+1} et les mots $\#0^{2^{n+1}} 1$ pour tout n tel que $M(n) \downarrow$. Le sous-shift X_M est donc constitué des mots biinfinis suivants :

$$\begin{aligned} & \dots \# \# 0^{2^{n+1}} 1 1 \dots \quad \text{avec } M(n) \uparrow \\ & \dots \# \# \# \# \dots \\ & \dots \# \# 0 0 0 \dots \\ & \dots 0 0 0 0 0 0 \dots \\ & \dots 1 1 1 1 1 1 \dots \\ & \dots 0 0 0 1 1 1 \dots \end{aligned}$$

- Y_M : on interdit les mots $\#1, 1\#, 10, 0\#$, les mots $\#0^k 1$ quand k n'est pas de la forme $2^{i+1} + 2^i$ et les mots $\#0^{2^{n+1}+2^n} 1$ pour tout n tel que $M(n) \downarrow$. Le sous-shift Y_M est donc constitué des mots biinfinis suivants :

$$\begin{aligned} & \dots \# \# 0^{2^{n+1}+2^n} 1 1 \dots \quad \text{avec } M(n) \uparrow \\ & \dots \# \# \# \# \dots \\ & \dots \# \# 0 0 0 \dots \\ & \dots 0 0 0 0 0 0 \dots \\ & \dots 1 1 1 1 1 1 \dots \\ & \dots 0 0 0 1 1 1 \dots \end{aligned}$$

Prouvons maintenant que X_M et Y_M sont conjugués si et seulement si l'ensemble $H_M = \{n \mid M(n) \uparrow\}$ est fini :

- ⇒ Si H_M est fini, alors il existe N qui majore tous ses éléments. Il existe alors clairement une fonction de conjugaison F avec un rayon $r_F > 2^{N+1} + 2^N$: celle-ci ne fait que décaler le début de la séquence infinie de 1.
- ⇐ Si H_M est infini, supposons qu'il existe une fonction de conjugaison $F : Y_M \rightarrow X_M$: premièrement, notons que $\#^{r_F}$, 0^{r_F} et 1^{r_F} doivent nécessairement avoir pour images $\#$, 0 et 1 respectivement. En effet, si ce n'est pas le cas, alors les mots de la forme $\dots \#0^k A \dots$ de X_M ne peuvent pas avoir de préimage. Prenons maintenant $n \in H_M$ tel que $2^n > 2r_F + 1$. Le point

$$\dots \# \# 0^{2^{n+1}+2^n} 11 \dots$$

a une image qui n'appartient pas au sous-shift X_M , car elle est de la forme

$$\dots \# \# w_1 \dots w_{2r} 0^{2^{n+1}+2^n-2r} w'_1 \dots w'_{2r} 11 \dots$$

avec $w_i, w'_i \in \{\#, 0, 1\}$. □

4.2 Factorisation

Nous pouvons voir assez rapidement que pour les SFTs, la factorisation est plus dure que la conjugaison, certains cas particuliers très simples étant déjà de complexité élevée :

Théorème 4.2.1

Soit Y le SFT contenant exactement une configuration, la configuration uniforme. Étant donné X en entrée, il est Π_1^0 -complet de savoir si X se factorise sur Y .

Preuve. La fonction de factorisation doit nécessairement envoyer tout symbole vers l'unique symbole de Σ_Y . Le problème est donc équivalent à savoir si X est *non-vide*, problème qui est Π_1^0 -complet. □

Théorème 4.2.2

Soit Y le SFT vide. Étant donné un SFT X en entrée, savoir si X se factorise sur Y est Σ_1^0 -complet.

Preuve. Cette fois-ci le problème est équivalent à savoir si X est vide, problème qui est Σ_1^0 -complet. □

Ces deux cas triviaux permettent déjà de savoir que la factorisation est nécessairement plus dure que la conjugaison. Mais on peut également se demander si pour tout degré $\deg_T d$, il existe un SFT X tel que le problème de la factorisation vers ce SFT soit de degré $\deg_T d$.

On s'intéresse maintenant au cas général, on va montrer ici que la factorisation est Σ_3^0 -complète.

4.2.1 La factorisation est Σ_3^0

Nous prouvons ici que la factorisation est dans Σ_3^0 quand on prend en entrée deux sous-shifts effectifs :

Théorème 4.2.3

Étant donnés deux sous-shifts effectifs X, Y en entrée, savoir si X se factorise sur Y est Σ_3^0 .

Preuve. X se factorise sur Y si et seulement s'il existe une fonction continue commutant avec le shift telle que $F(X) = Y$. Deviner F est le premier quantificateur existentiel. Le lemme suivant et le lemme 4.1.6 permettent de conclure la preuve. \square

Lemme 4.2.4

Étant donnés deux sous-shifts effectifs X, Y et une fonction continue commutant avec le shift F , décider si $Y \subseteq F(X)$ est Π_2^0 .

Preuve. On prouve que $Y \subseteq F(X)$ est équivalent à la proposition suivante :

Pour tout motif extensible m de Y , $F^{-1}(m)$ contient un motif admissible.

Cette proposition est Π_2^0 car vérifier qu'un motif n'est pas extensible est Σ_1^0 , et vérifier qu'il existe un motif admissible dans $F^{-1}(m)$ est Π_1^0 .

Prouvons maintenant l'équivalence :

- Si $Y \subseteq F(X)$ alors tout motif extensible m de Y apparaît dans un point $y \in Y$ qui a une préimage $x \in X$, donc m a une préimage extensible.
- Inversement, supposons que tout motif extensible m de Y a une préimage admissible. Soit y un point de Y , il existe alors une suite croissante m_i de motifs extensibles convergeant vers y . Tous ces motifs ont au moins une préimage admissible m'_i par F . Par compacité, on peut extraire de m'_i une suite convergeant vers un point $x \in X$. Par construction x est une préimage de y , donc $Y \subseteq F(X)$. \square

4.2.2 La factorisation est Σ_3^0 -dure

Nous allons prouver ici que la factorisation, quand on prend en entrée deux SFTs, est Σ_3^0 -dure. Afin de faire cela, nous allons utiliser le jeu de tuiles T introduit au chapitre 2 en section 2.2.1. Le point α de T contient une grille \mathbb{N}^2 dont les frontières (positions $(i, 0)$ et $(0, j)$) sont différenciables des autres positions. Dans le chapitre 2, on a vu que l'on pouvait superposer sur cette grille n'importe quel langage bidimensionnel vérifiable par contraintes locales. On dira qu'un SFT résultant d'une telle superposition a une T -structure.

Définition 4.2.5*T-structure*

Un SFT X a une T -structure si c'est une copie de T sur laquelle on a ajouté de nouveaux symboles et règles, mais uniquement sur la grille. Les configurations contenant un coin, et donc une grille sont appelées α -configurations.

Un SFT avec une T -structure peut ne pas avoir d' α -configuration : en effet, si les symboles superposés à la grille ne "paivent" pas \mathbb{N}^2 , alors il ne restera que des points ayant comme base les configurations surnuméraires (avec au plus une ligne horizontale) que l'on a listé en figure 2.4.

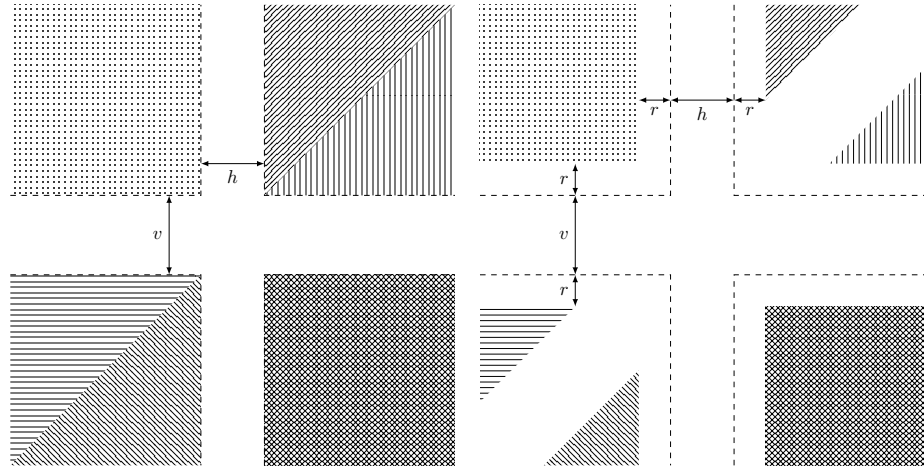
L'intérêt des SFTs avec une T -structure est que pour deux SFTs X, Y ayant chacun une T -structure, la structure est préservée par préimage par fonction locale (quand il y a une préimage).

Lemme 4.2.6

Soient X, Y deux SFTs avec une T -structure et F une fonction continue commutant avec le shift de rayon r . Alors tout point α de Y ayant une préimage, a pour préimage un point α , dont le coin est translaté par \mathbf{v} avec $\|\mathbf{v}\|_\infty \leq r$.

Preuve. D'après le lemme 2.2.2, on sait que les points de T autres que α , ceux de la figure 2.4, ont au plus une ligne horizontale et deux verticales (de part et d'autre d'une horizontale dans ce cas). Ces points sont donc constitués de deux quarts de plan uniformes et de quatre huitièmes de plan uniformes avec une zone non uniforme au centre, voir figure 4.1a. L'image de ces points est donc nécessairement de la même forme, voir figure 4.1b : deux quarts de plan uniformes et quatre huitièmes de plan uniformes. Les α -configurations n'étant pas de cette forme, elles ne peuvent pas être image de l'une de ces configurations.

Il faut maintenant prouver la deuxième partie : la structure d'une α -configuration ne peut pas être translatée par un vecteur \mathbf{v} tel que $\|\mathbf{v}\|_\infty > r$. Supposons qu'une α -configuration $x \in X$ se factorise sur une α -configuration $y \in Y$ et déplace la structure de $\mathbf{v} = (\mathbf{v}_x, \mathbf{v}_y)$, avec $\|\mathbf{v}\|_\infty > r$. Sans perte de généralité, on peut supposer que $\mathbf{v}_x > r$ et $\mathbf{v}_y > 0$ et que le coin de x est en $(0, 1)$: pour tout $k \in \mathbb{N}^*$, il y a un carré commençant à la position $(2k^2 + k, 2k^2 + k)$, dans ce carré il y a deux carrés $(k - 1) \times (k - 1)$ uniformes, voir figure 4.2. Prenons $k > (\|\mathbf{v}\|_\infty + r + 2)$, alors par hypothèse il y a un symbole t de ligne verticale en position $\mathbf{z}_x = (2k^2 + k, 2k^2 + 2k + 1)$ dans x , et donc en position $\mathbf{z}_y = (2k^2 + k + \mathbf{v}_x, 2k^2 + 2k + 1 + \mathbf{v}_y)$ dans y . On sait que $x|_{\mathbf{z}_y + \mathcal{B}_r}$ a comme image t . Mais on a vu précédemment que $x|_{\mathbf{z}_y + \mathcal{B}_r} = x|_{\mathbf{z}_y + (1, 0) + \mathcal{B}_r}$ car ils sont uniformes, on devrait donc avoir deux symboles t voisins sur y aux positions \mathbf{z}_y et $\mathbf{z}_y + (1, 0)$, ce qui est impossible. \square



(a) Les quarts et huitièmes de plan uniformes (b) L'image par F de ces morceaux de plans des configurations surnuméraires de T . Les uniformes est toujours uniforme. On notera distances horizontale h et verticale v entre qu'elle est décalée de r , le rayon de F . ces plans correspondent à l'éloignement des lignes/signaux.

FIGURE 4.1 – Les images des configurations surnuméraires sont nécessairement des configurations surnuméraires. Les zones blanches sont les zones qui ne sont pas uniformes.

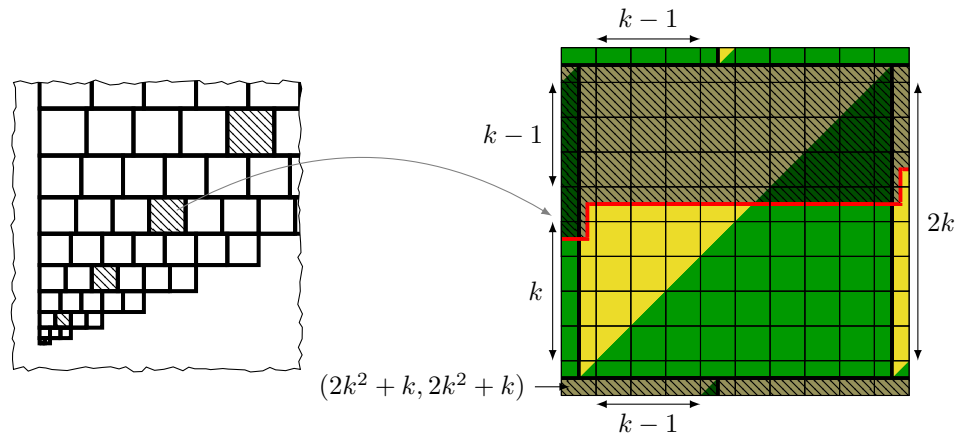


FIGURE 4.2 – Pour tout $k \in \mathbb{N}^*$, le carré commençant à la position $(2k^2 + k, 2k^2 + k)$ est de la forme à droite. On peut voir deux carrés $(k-1) \times (k-1)$ uniformes aux positions $(2k^2 + k + 1, 2k^2 + 2k + 2)$ et $(2k^2 + 2k + 2, 2k^2 + k + 1)$.

Avant de nous attaquer à la réduction pour les SFTs, nous donnons une réduction dans le cas des sous-shifts effectifs de dimension 1 qui utilise la même idée :

Théorème 4.2.7

Étant donnés deux sous-shifts effectifs X, Y en entrée, il est Σ_3^0 -dur de décider si X se factorise sur Y .

Preuve. On va effectuer une réduction à partir de COFIN, l'ensemble des machines de Turing qui ne s'arrêtent pas sur un nombre fini d'entrées, voir le chapitre 1, section 1.2.2.

Étant donné une machine de Turing M , nous construisons deux sous-shifts effectifs X_M et Y_M tels que X_M se factorise sur Y_M si et seulement si l'ensemble des entrées sur lesquelles M s'arrête est fini :

- X_M a pour alphabet $\{\#, B, R, 0, 1, @\}$ et est constitué des points suivants :

$$\begin{aligned} & \dots \#\#B^{n+1}0@@ \dots \quad \text{avec } M(n) \uparrow \\ & \dots \#\#B^{n+1}1@@ \dots \quad \text{avec } M(n) \uparrow \\ & \dots BBBb@@@ \dots \quad \text{avec } b \in \{0, 1\} \\ & \dots \#\#\#BBB \dots \\ & \dots \#\#\#RRR \dots \\ & \dots \#\#\#\#\# \dots \\ & \dots BBBBBB \dots \\ & \dots RRRRRR \dots \end{aligned}$$

X_M est un sous-shift effectif car les seuls motifs interdits "complexes" à énumérer sont ceux de la forme $\#B^n b @$ où $b \in \{0, 1\}$ avec $M(n) \downarrow$ et les autres sont les motifs à deux symboles suivants :

$$\begin{aligned} & 0\#, 1\#, B\#, @\#, R\#, \\ & 0B, 1B, @B, RB, \\ & \#R, BR, 0R, 1R, @R, \\ & \#0, R0, 10, @0, \\ & \#1, R1, 01, @1, \\ & \#@, B@, R@ \end{aligned}$$

- Y_M a pour alphabet $\{\#, B, 0, 1, @\}$ et est constitué des points suivants :

$$\begin{aligned} & \dots \#\#0B^{n+1}@@ \dots \quad \text{avec } M(n) \uparrow \\ & \dots \#\#1B^{n+1}@@ \dots \quad \text{avec } M(n) \uparrow \\ & \dots \#\#\#bBBB \dots \quad \text{avec } b \in \{0, 1\} \\ & \dots \#\#\#@@@ \dots \\ & \dots \#\#\#\#\# \dots \\ & \dots BBBBBB \dots \end{aligned}$$

Il est clair que Y_M est aussi un sous-shift effectif.

On appelle le symbole $\{0, 1\}$ apparaissant dans certains points de X_M et Y_M la *décoration*. Vérifions que X_M se factorise sur Y_M si et seulement si l'ensemble des entrées sur lesquelles M ne s'arrête pas est fini :

- Quand le nombre d'entrées sur lesquelles M ne s'arrête pas est fini il existe N majorant toutes les entrées. On peut prendre une fonction de factorisation de taille $N + 3$, qui déplace la décoration qui est à la fin vers le début. Et qui transforme la configuration $\dots \#\#\#BBB\dots$ en $\dots \#\#\#0BBB\dots$ et la configuration $\dots \#\#\#RRR\dots$ en $\dots \#\#\#1BBB\dots$, on peut déduire les images des autres points aisément. Notons que l'on a été obligé de rajouter la configuration $\dots \#\#\#RRR\dots$ dans X_M afin de palier à la décoration manquante quand le mot $B\dots B$ devient infini.
- Supposons que le nombre d'entrée sur lesquelles M ne s'arrête pas est infini et qu'il existe une fonction de factorisation F de X_M vers Y_M de rayon r . Comme il y a des points de la forme $\dots \#\#\#bB^{n+1}@@@ \dots$ pour n arbitrairement grand dans Y_M , les r -blocs $\#\dots\#$, $B\dots B$ et $@\dots@$ ont nécessairement pour images $\#$, B et $@$. Mais comme la fonction de factorisation F est de rayon r , alors pour tout $n > r$ tel que $M(n) \uparrow$, les points $\dots \#\#B^{n+1}0@@ \dots$ et $\dots \#\#B^{n+1}1@@ \dots$ de X_M ont nécessairement la même image : $\dots \#\#bB^{n+1}@@ \dots$. Les points $\dots \#\#\bar{b}B^{n+1}@@ \dots$ de Y_M n'ont donc pas de préimage dans X_M et F ne peut pas être une fonction de factorisation. □

L'idée de la preuve pour les SFTs de dimension $d \geq 2$ est maintenant la même, mais on doit cette fois-ci coder explicitement le calcul dans le SFT : on utilise un SFT avec une T -structure pour cela.

Théorème 4.2.8

Étant donné deux SFTs X, Y en entrée, il est Σ_3^0 -dur de décider si X se factorise sur Y .

Preuve. La réduction s'effectue à nouveau à partir de COFIN, l'ensemble des machines de Turing qui ne s'arrêtent pas sur un nombre fini d'entrées.

Étant donné une machine de Turing M , nous construisons deux SFTs X_M et Y_M tels que X_M se factorise sur Y_M si et seulement si l'ensemble des entrées sur lesquelles M ne s'arrête pas est fini.

Tout d'abord, construisons un SFT Z_M ayant une T -structure, nous nous en servons comme base pour construire X_M et Y_M . On prend donc le SFT T comme base. On ajoute deux symboles, un blanc et un bleu, par dessus la grille, avec les règles suivantes :

- Une couleur se transmet horizontalement à ses voisins de gauche et de droite, de manière à ce que toutes les intersections de la grille à chaque hauteur soient de la même couleur.
- La couleur bleue se transmet verticalement aux intersections de la grille du niveau du dessus. Ainsi lorsque qu'une ligne horizontale est d'une couleur, toutes les lignes au dessus sont également bleues.

À ce stade, pour chaque $n \in \mathbb{N} \cup \{\infty\}$, à translation près, on a une configuration avec une grille divisée en deux parties, une partie bleue et une partie blanche. La première ligne bleue étant la n -ième. De plus cette première ligne bleue a exactement n intersections, une vue symbolique est présentée sur la figure 4.3.

On ajoute maintenant du calcul sur la partie bleue uniquement, la largeur de la

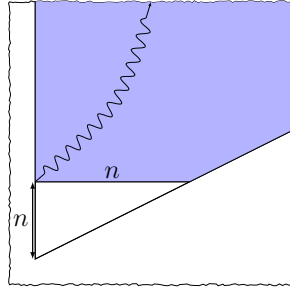


FIGURE 4.3 – Les zones bleues permettent de coder du calcul sur les configurations avec grilles : le nombre de lignes horizontales entre le coin et le début de la zone bleue correspond à la taille de l'entrée du calcul.

première ligne de calcul constitue l'entrée. Pour l'entrée n , le calcul est donc distant de $n(n+1)/2$ de la première ligne, celle qui ne contient que le coin.

À ce stade, on a une grille pour chaque $n \in \mathbb{N}$ tel que $M(n) \uparrow$, le calcul commençant à la n -ième ligne. Si $M(n) \downarrow$, alors il n'y a pas de point α dont la première ligne de calcul contient n intersections. Si par exemple M est totale, alors il n'y a qu'un seul point α , celui qui ne contient pas de zone bleue.

On a fini la construction de Z_M , on donne maintenant X_M et Y_M :

- X_M : On construit d'abord un SFT Z'_M en ajoutant une décoration, 0 ou 1, à l'intersection la plus à gauche de la première ligne bleue uniquement. X_M est Z'_M auquel on ajoute une troisième couleur, rouge, qui ne peut apparaître que toute seule, à la place du blanc et du bleu. On ne met pas de calcul sur la couleur rouge. On a en fait ajouté un deuxième point α sans calcul, en plus de celui qui est entièrement blanc.
- Y_M : est obtenu à partir de Z_M en ajoutant simplement une décoration, 0 ou 1, sur le symbole de coin uniquement.

Vérifions maintenant que X_M se factorise sur Y_M si et seulement si l'ensemble des entiers sur lesquels M ne s'arrête pas est fini :

- Si le nombre d'entrées sur lesquelles M ne s'arrête pas est fini, alors il existe N , tel que pour tout $n > N$, $M(n) \downarrow$. La fonction de factorisation suivante fonctionne :
 - F est l'identité sur la composante Z_M .
 - F a un rayon $r > N^2 + N$, c'est à dire que pour la plus grande entrée sur laquelle M ne s'arrête pas, la fonction de factorisation centrée sur le coin a dans sa fenêtre le début du calcul de M , et donc la décoration.
 - La décoration du point d'arrivée est la même que celle de sa préimage. Le cas particulier des deux configurations sans calcul est réglé grâce à la couleur rouge de X_M : s'il n'y a pas de calcul dans la fenêtre et que le coin est blanc dans la préimage, alors le coin de l'image a comme décoration 0, s'il est rouge il a comme décoration 1.

On peut noter que la réduction est toujours valide même quand M est totale.

- Inversement, supposons que l'ensemble des entrées sur lesquelles M ne s'arrête pas soit infini et qu'il existe une fonction de factorisation F de rayon r factorisant X_M vers Y_M . D'après le lemme 4.2.6, un point α de Y_M a nécessairement comme préimage un point α de X_M , de plus le coin de la préimage est dans la fenêtre de celui de l'image. Il n'y a qu'une seule fenêtre

\mathcal{B}_r , centrée sur le coin, apparaissant dans une infinité de points α de X_M , celle où le calcul n'apparaît pas (encore) : celle où toutes les intersections sont blanches. Il ne peut donc pas y avoir une infinité d'images de points α ayant 0 et 1 comme décorations sur leurs coins : il ne peut y avoir qu'une infinité de points avec 0 ou une infinité de points avec 1, mais pas les deux. Certains points de Y_M n'ont donc pas de préimage.

□

Remarque 4.2.1. *Les deux SFTs ne peuvent jamais être conjugués : les configurations surnuméraires dont les lignes sont rouges et celles dont les lignes sont blanches ont les mêmes images.*

On a ainsi terminé la preuve du théorème 4.0.5 qui consiste en les théorèmes 4.2.3 et 4.2.8.

Chapitre 5

Bibliographie

5.1 Bibliographie personnelle

- [JV10a] E. Jeandel et P. Vanier. « Periodicity in Tilings ». Dans : *Developments in Language Theory*. T. 6224. Lecture Notes in Computer Science. Springer, 2010, p. 243–254. ISBN : 978-3-642-14454-7.
- [JV10b] E. Jeandel et P. Vanier. « Slopes of Tilings ». Dans : *JAC*. Turku Center for Computer Science, 2010, p. 145–155. ISBN : 978-952-12-2503-1.
- [JV11] E. Jeandel et P. Vanier. « Π_1^0 Sets and Tilings ». Dans : *TAMC*. T. 6648. Lecture Notes in Computer Science. Springer, 2011, p. 230–239. ISBN : 978-3-642-20876-8.
- [JV12a] E. Jeandel et P. Vanier. « Hardness of conjugacy and factorization of multidimensional subshifts of finite type ». Dans : *Preprint abs/1204.4988* (2012).
- [JV12b] E. Jeandel et P. Vanier. « Turing degrees of multidimensional SFTs ». Dans : *Accepté à Theoretical Computer Science* (2012).
- [KVZ09] J. Kari, P. Vanier et T. Zeume. « Bounds on Non-surjective Cellular Automata ». Dans : *MFCS*. T. 5734. Lecture Notes in Computer Science. Springer, 2009, p. 439–450. ISBN : 978-3-642-03815-0.

5.2 Bibliographie

- [AB09] N. Aubrun et M.-P. Béal. « Decidability of Conjugacy of Tree-Shifts of Finite Type ». Dans : *ICALP '09: Proceedings of the 36th International Colloquium on Automata, Languages and Programming*. Springer-Verlag, 2009, p. 132–143. URL : <http://monge.univ-mlv.fr/~aubrun/articles/icalp09.pdf>.
- [AD01] C. Allauzen et B. Durand. « Tiling Problems ». Dans : *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer-Verlag Berlin, Heidelberg, 2001. Chap. A, p. 407–420.

- [AS09] N. Aubrun et M. Sablik. « An Order on Sets of Tilings Corresponding to an Order on Languages ». Dans : *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*. 2009, p. 99–110.
- [BDG88] J. Balcazar, J. Diaz et J. Gabarro. *Structural complexity*. EATCS monographs on theoretical computer science v. 1. Springer-Verlag, 1988. ISBN : 9783540186229.
- [BDG90] J. Balcazar, J. Diaz et J. Gabarro. *Structural complexity*. Structural Complexity vol. 2. Springer-Verlag, 1990. ISBN : 9783540520795.
- [Ber64] R. Berger. « The Undecidability of the Domino Problem ». Thèse de doct. Harvard University, 1964.
- [Ber66] R. Berger. *The Undecidability of the Domino Problem*. 66. The American Mathematical Society, 1966.
- [Ber95] J. Berstel. « Axel Thue’s papers on repetitions in words: a translation ». Dans : *Publications du LaCIM 20* (1995).
- [Bir12] M.-G. D. Birkhoff. « Quelques théorèmes sur le mouvement des systèmes dynamiques ». Dans : *Bulletin de la SMF* 40 (1912), p. 305–323.
- [Bor08] B. Borchert. « Formal Language characterizations of P, NP, and PSPACE ». Dans : *Journal of Automata, Languages and Combinatorics* 13.3/4 (2008), p. 161–183.
- [Boy08] M. Boyle. « Open Problems in Symbolic Dynamics ». Dans : *Contemporary Mathematics* 469 (2008), p. 69–118.
- [Boy83] M. Boyle. « Lower entropy factors of sofic systems ». Dans : *Ergodic Theory and Dynamical Systems* 3 (1983), p. 541–551.
- [CDK08] D. Cenzer, A. Dashti et J. L. F. King. « Computable symbolic dynamics ». Dans : *Mathematical Logic Quarterly* 54.5 (2008), p. 460–469. DOI : 10.1002/ma1q.200710066.
- [Cen+10] D. Cenzer, A. Dashti, F. Toska et S. Wyman. « Computability of Countable Subshifts ». Dans : *Computability in Europe (CiE)*. T. 6158. Lecture Notes in Computer Science. 2010, p. 88–97. DOI : 10.1007/978-3-642-13962-8_10.
- [Cen+12] D. Cenzer, A. Dashti, F. Toska et S. Wyman. « Computability of Countable Subshifts in One Dimension ». Dans : *Theory of Computing Systems* (2012). DOI : 10.1007/s00224-011-9358-z.
- [CR] D. Cenzer et J. B. Remmel. *Effectively Closed Sets*. ASL Lecture Notes in Logic.
- [CR98] D. Cenzer et J. Remmel. « Π_1^0 classes in mathematics ». Dans : *Handbook of Recursive Mathematics - Volume 2: Recursive Algebra, Analysis and Combinatorics*. T. 139. Studies in Logic and the Foundations of Mathematics. Elsevier, 1998. Chap. 13, p. 623–821. DOI : 10.1016/S0049-237X(98)80046-3.

- [Cul96] K. Culik II. « An aperiodic set of 13 Wang tiles ». Dans : *Discrete Mathematics* 160.1–3 (1996), p. 245–251. ISSN : 0012-365X. DOI : 10.1016/S0012-365X(96)00118-5. URL : <http://www.sciencedirect.com/science/article/pii/S0012365X96001185>.
- [DLS08] B. Durand, L. A. Levin et A. Shen. « Complex tilings. » Dans : *Journal of Symbolic Logic* 73.2 (2008), p. 593–613.
- [DRS09] B. Durand, A. E. Romashchenko et A. Shen. « Fixed Point Theorem and Aperiodic Tilings ». Dans : *Bulletin of the EATCS* 97 (2009), p. 126–136.
- [DRS10] B. Durand, A. Romashchenko et A. Shen. « Effective Closed Subshifts in 1D Can Be Implemented in 2D ». Dans : *Fields of Logic and Computation*. Lecture Notes in Computer Science 6300. Springer, 2010, p. 208–226. DOI : 10.1007/978-3-642-15025-8_12.
- [DRS12] B. Durand, A. Romashchenko et A. Shen. « Fixed-point tile sets and their applications ». Dans : *Journal of Computer and System Sciences* 78.3 (mai 2012), p. 731–764. DOI : 10.1016/j.jcss.2011.11.001.
- [Dur99] B. Durand. « Tilings and Quasiperiodicity ». Dans : *Theoretical Computer Science* 221.1-2 (juin 1999), p. 61–75.
- [Emd97] P. van Emde Boas. « The convenience of Tilings ». Dans : *Complexity, Logic, and Recursion Theory*. T. 187. Lecture Notes in Pure and Applied Mathematics. CRC, 1997.
- [Han74] W. Hanf. « Non Recursive Tilings of the Plane I ». Dans : *Journal of Symbolic Logic* 39.2 (juin 1974), p. 283–285.
- [Har84] D. Harel. « A Simple Highly Undecidable Domino Problem ». Dans : *Proc. Conf. on Logic and Computation* (1984).
- [Har85] D. Harel. « Recurring Dominoes: Making the Highly Undecidable Highly Understandable ». Dans : *Annals of Discrete Mathematics* 24 (1985), p. 51–72.
- [Hed69] G. A. Hedlund. « Endomorphisms and automorphisms of the shift dynamical system ». Dans : *Theory of Computing Systems* 3.4 (1969), p. 320–375.
- [Hin12] P. G. Hinman. « A survey of Muchnik and Medvedev degrees ». Dans : *Bulletin of Symbolic Logic* 18.2 (2012), p. 161–229.
- [HM10] M. Hochman et T. Meyerovitch. « A characterization of the entropies of multidimensional shifts of finite type ». Dans : *Annals of Mathematics* 171.3 (mai 2010), p. 2011–2038. DOI : 10.4007/annals.2010.171.2011.
- [Hoc09a] M. Hochman. « A note on universality in multidimensional symbolic dynamics ». Dans : *Discrete and Continuous Dynamical Systems S* 2.2 (2009). DOI : 10.3934/dcdss.2009.2.301.
- [Hoc09b] M. Hochman. « On the dynamics and recursive properties of multidimensional symbolic systems ». Dans : *Inventiones Mathematicae* 176.1 (avr. 2009), p. 2009.

- [Imm88] N. Immerman. « Nondeterministic space is closed under complementation ». Dans : *SIAM Journal on Computing* 17.5 (oct. 1988), p. 935–938.
- [JS72] C. G. Jockusch et R. I. Soare. « \prod_1^0 Classes and Degrees of Theories ». Dans : *Transactions of the American Mathematical Society* 173 (nov. 1972), p. 33–56.
- [JS74] N. D. Jones et A. L. Selman. « Turing machines and the spectra of first-order formulas ». Dans : *Journal of Symbolic Logic* 39.1 (mar. 1974), p. 139–150.
- [Kar90a] J. Kari. « Decision Problems concerning Cellular Automata ». Thèse de doct. University of Turku, 1990.
- [Kar90b] J. Kari. « Reversibility of 2D cellular automata is undecidable ». Dans : *Physica D: Nonlinear Phenomena* 45.1-3 (1990), p. 379–385.
- [Kar92] J. Kari. « The Nilpotency Problem of One-Dimensional Cellular Automata ». Dans : *SIAM Journal on Computing* 21.3 (1992), p. 571–586.
- [Kar94] J. Kari. « Reversibility and surjectivity problems of cellular automata ». Dans : *Journal of Computer and System Sciences* 48.1 (1994), p. 149–182. DOI : 10.1016/S0022-0000(05)80025-X.
- [Kar96] J. Kari. « A small aperiodic set of Wang tiles ». Dans : *Discrete Mathematics* 160 (1996), p. 259–264.
- [Kec95] A. S. Kechris. *Classical descriptive set theory*. T. 156. Graduate Texts in Mathematics. New York : Springer-Verlag, 1995, p. xviii+402. ISBN : 0-387-94374-9.
- [KMW62] A. Kahr, E. F. Moore et H. Wang. « Entscheidungsproblem reduced to the $\forall\exists\forall$ case ». Dans : *Proceedings of the National Academy of Sciences of the United States of America* 48.3 (mar. 1962), p. 365–377.
- [Knu05] D. E. Knuth. « The Art of Computer Programming ». Dans : t. 4 fasc. 2. Addison-Wesley, 2005. Chap. Generating all Tuples and Permutations.
- [KOR00] K. H. Kim, N. S. Ormes et F. W. Roush. « The spectra of nonnegative integer matrices via formal power series ». Dans : *Journal of the American Mathematical Society* 13 (2000), p. 773–806.
- [Koz06] D. Kozen. *Theory of Computation*. New York : Springer, 2006. ISBN : 13: 978-1-84628-297-3.
- [Kur66] K. Kuratowski. *Topology, Vol. I, 3rd edition*. NY: Academic Press, 1966.
- [Lev73] L. A. Levin. « Universal Search Problems ». Dans : *Problemy Peredachi Informatsii* 9.3 (juil. 1973). traduction dans [Tra84], p. 115–116.
- [Lev84] L. A. Levin. « Problems, Complete in “Average” Instance ». Dans : *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 1984, p. 465. DOI : 10.1145/800057.808713.

- [Lev86] L. A. Levin. « Average Case Complete Problems ». Dans : *SIAM Journal on Computing* 15.1 (1986), p. 285–286. DOI : 10.1137/0215020.
- [Lin04] D. A. Lind. « Multi-Dimensional Symbolic Dynamics ». Dans : *Symbolic Dynamics and its Applications*. Sous la dir. de S. G. Williams. Proceedings of Symposia in Applied Mathematics 60. American Mathematical Society. 2004, p. 61–79.
- [Lin84] D. A. Lind. « The entropies of topological Markov shifts and a related class of algebraic integers ». Dans : *Ergodic Theory and Dynamical Systems* 4.02 (1984), p. 283–300. DOI : 10.1017/S0143385700002443. eprint : http://journals.cambridge.org/article_S0143385700002443. URL : <http://dx.doi.org/10.1017/S0143385700002443>.
- [LM95] D. A. Lind et B. Marcus. *An Introduction to Symbolic Dynamics and Coding*. New York, NY, USA : Cambridge University Press, 1995. ISBN : 0521551242.
- [Med55] Y. T. Medvedev. « Degrees of difficulty of the mass problem ». Dans : *Doklady Akademii Nauk SSSR* 104 (1955), 501–504.
- [Mey10] T. Meyerovitch. « Growth-type invariants for \mathbb{Z}^d subshifts of finite type and arithmetical classes of real numbers ». Dans : *Inventiones Mathematicae* 184.3 (2010). DOI : 10.1007/s00222-010-0296-1.
- [MH38] M. Morse et G. A. Hedlund. « Symbolic Dynamics ». Dans : *American Journal of Mathematics* 60.4 (oct. 1938), p. 815–866.
- [Mil12] J. S. Miller. « Two Notes on subshifts ». Dans : *Proceedings of the American Mathematical Society* 140.5 (2012), p. 1617–1622. DOI : 10.1090/S0002-9939-2011-11000-1.
- [Mor21] H. M. Morse. « Recurrent Geodesics on a Surface of Negative Curvature ». Dans : *Transactions of the American Mathematical Society* 22.1 (jan. 1921), p. 84–100. DOI : 10.1090/S0002-9947-1921-1501161-8.
- [Muc63] A. A. Muchnik. « On strong and weak reducibility of algorithmic problems (traduction) ». Dans : *Sibirskii Matematicheskii Zhurnal* 4 (1963), 1328–1341.
- [Mye74] D. Myers. « Non Recursive Tilings of the Plane II ». Dans : *Journal of Symbolic Logic* 39.2 (juin 1974), p. 286–294.
- [Nas95] M. Nasu. « Textile systems for endomorphisms and automorphisms of the shift ». Dans : *Memoirs of the American Mathematical Society* 546 (1995).
- [Pou10] V. Poupet. « Yet Another Aperiodic Tile Set ». Dans : *JAC*. Sous la dir. de J. Kari. Turku Center for Computer Science, 2010, p. 191–202. ISBN : 978-952-12-2503-1.
- [Rob71] R. M. Robinson. « Undecidability and Nonperiodicity for Tilings of the Plane ». Dans : *Inventiones Math.* 12 (1971).
- [Rog87] H. Rogers. *Theory of Recursive Functions and Effective Computability*. MIT Press Cambridge, MA, USA, 1987.

- [Sha48] C. E. Shannon. « A mathematical theory of communication ». Dans : *Bell System Technical Journal* 27 (1948), p. 379–423.
- [Sim11a] S. G. Simpson. « Mass problems associated with effectively closed sets ». Dans : *Tohoku Mathematical Journal* 63.4 (2011), p. 489–517.
- [Sim11b] S. G. Simpson. « Medvedev Degrees of 2-Dimensional Subshifts of Finite Type ». Dans : *Ergodic Theory and Dynamical Systems* (2011).
- [Sze88] R. Szelepcsényi. « The method of forced enumeration for nondeterministic automata ». Dans : *Acta Informatica* 26 (3 1988). 10.1007/BF00299636, p. 279–284. ISSN : 0001-5903. URL : <http://dx.doi.org/10.1007/BF00299636>.
- [Thu12] A. Thue. « Über die gegenseitige Lage gleicher Teiler gewisser Zeichenreihen ». Dans : *Kra. Vidensk. Selsk. Skrifter. I. Mat.-Nat. Kl.* 12 (1912). traduction dans [Ber95].
- [Tra84] B. A. Trakhtenbrot. « A survey of Russian Approached to *Perebor* (Brute-Force Search) Algorithms ». Dans : *Annals of the History of Computing* 6.4 (oct. 1984), p. 384–400. DOI : 10.1109/MAHC.1984.10036.
- [Wan60] H. Wang. « Proving Theorems by Pattern Recognition I ». Dans : *Communications of the ACM* 3.4 (avr. 1960), p. 220–234.
- [Wan61] H. Wang. « Proving theorems by Pattern Recognition II ». Dans : *Bell Systems technical journal* 40 (1961), p. 1–41.
- [Wei73] B. Weiss. « Subshifts of finite type and sofic systems ». Dans : *Monatshefte für Mathematik* 77 (1973), p. 462–474.
- [Wil73] R. F. Williams. « Classification of subshifts of finite type ». Dans : *Annals of Mathematics* 98 (1973), p. 120–153.
- [ZW01] X. Zheng et K. Weihrauch. « The Arithmetical Hierarchy of Real Numbers ». Dans : *Mathematical Logic Quarterly* 47.1 (2001), p. 51–65. ISSN : 1521-3870. DOI : 10.1002/1521-3870(200101)47:1<51::AID-MALQ51>3.0.CO;2-W. URL : [http://dx.doi.org/10.1002/1521-3870\(200101\)47:1<51::AID-MALQ51>3.0.CO;2-W](http://dx.doi.org/10.1002/1521-3870(200101)47:1<51::AID-MALQ51>3.0.CO;2-W).

Index

- T -structure, 91
- d -cubes, 63
- \mathcal{X}_τ , 8
- Σ_n^0 -complet, 24
- Σ_n^0 -dur, 24
- alphabet, 3
- bloc
 - n -bloc, 4
- bloc, 4
- classe Π_1^0 , 35
- classification de sous-shifts, 20
- co-récursivement énumérable, 23
- composante, 9
- configuration
 - minimale, 19
 - quasipériodique, 19
 - strictement quasipériodique, 19
- configuration, 3
- conjugaison, 10
- couche, 9
- déterminisme
 - Est, 62
 - Nord-Ouest, 65
- degré
 - many-one, 23
 - Medvedev, 36
 - Muchnik, 36
 - Turing, 23
- ensemble Σ_n^0 , Π_n^0 , Δ_n^0 , 24
- ensembles effectivement clos, 35
- entropie, 20
- espace de pavages, 7
- espace des configurations, 3
- facteur, 11
- factorisation, 11
- faisceau, 48
- fenetre, 19
- fonction
 - calculable, 22
 - récursive, 22
 - récursive totale, 23
- full-shift, 3
- invariant de conjugaison, 20
- jeu de tuiles, 7
- jump Turing, 23
- motif
 - admissible, 4, 84
 - autorisé, 6
 - extensible, 4, 84
- motif, 4
- orbite, 16
- période
 - forte, 55
- période, 55
- périodique
 - 1-périodique, 53
 - k -périodique, 55
 - fortement, 53, 55
 - horizontalement, 53, 55
- périodique, 55
- pavage, 7
- pavage valide, 7
- plongement, 11
- points, 4
- problème, 23
- problèmes de masse, 35
- récursivement énumérable, 23
- récursivement isomorphe, 23
- réduction
 - many-one, 23
 - Turing, 23
- rayon

- fonction continue, 10
- sous-shift, 6

- shift-invariance, 4
- shift-invariant, 4
- sous-shift, 4
 - apériodique, 18
 - de type fini, 6
 - dimension, 4
 - effectif, 13
 - minimal, 19
 - sofique, 11
- superposition de sous-shifts, 8
- support, 4
- symboles, 3

- Tuile de Wang, 7

- union disjointe de sous-shifts, 8

