



**HAL**  
open science

# Structural and computational complexity of tilings and cellular automata

Julien Cervelle

► **To cite this version:**

Julien Cervelle. Structural and computational complexity of tilings and cellular automata. Computer Science [cs]. Université de provence, 2002. English. NNT : . tel-01208375

**HAL Id: tel-01208375**

**<https://hal.science/tel-01208375v1>**

Submitted on 14 Oct 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC0 - Public Domain Dedication 4.0 International License

n° attribué par la bibliothèque :

Année 2002

# THÈSE

*présentée à*

**L'UNIVERSITÉ DE PROVENCE**

*au sein du*

Laboratoire d'Informatique Fondamentale de Marseille

*pour obtenir le titre de*

**DOCTEUR DE L'UNIVERSITÉ DE PROVENCE**

Discipline : **Informatique**

École Doctorale de Mathématique et d'Informatique de Marseille

*présentée et soutenue publiquement par*

**Julien CERVELLE**

le 4 janvier 2002

## Complexité structurelle et algorithmique des pavages et des automates cellulaires

Devant le jury composé de :

Alain COLMERAUER	Professeur, Université de la Méditerranée	<i>président</i>
Marianne DELORME	Maître de Conférences, ÉNS de Lyon	
Bruno DURAND	Professeur, Université de Provence	<i>directeur</i>
Enrico FORMENTI	Maître de Conférences, Université de Provence	<i>co-directeur</i>
Jean-Yves MARION	Maître de Conférences, Université Nancy II	<i>rapporteur</i>
Michel MORVAN	Professeur, Université Paris VII	<i>rapporteur</i>



# Remerciements

Je remercie tout d'abord vivement mon directeur de thèse. C'est en effet lui qui m'a guidé lors de ma scolarité à l'École Normale Supérieure de Lyon, alors qu'il était mon tuteur. Il a notamment su me faire découvrir certains aspects de l'informatique et me détourner des mathématiques, malgré bon nombre de mises en garde concernant le peu de moyen des théoriciens en informatique. Enfin, durant ma thèse, il a su me guider dans mes recherches tout en me laissant une liberté d'initiative.

Je remercie aussi Enrico Formenti, non seulement pour son aide dans mes travaux sur les automates cellulaires : c'est par son intermédiaire que j'ai appris la plupart des résultats de ce domaine, mais aussi pour son omniprésence, et sa réussite à me supporter dans son bureau depuis déjà deux ans.

Je suis très reconnaissant envers les membres du jury d'avoir bien voulu juger mon travail, de l'avoir lu complètement, de s'être déplacé si proche du réveillon de la nouvelle année, et d'avoir su poser des questions pertinentes, qui m'ont permis de réfléchir un peu plus à la suite de mes recherches. Les rapporteurs, Jean-Yves Marion et Michel Morvan, pour bien avoir cerné les qualités et les défauts de mon mémoire, reçoivent toute ma gratitude.

Toute l'équipe du LIF, de MC2, et de l'ENS en général a, dans la bonne humeur, constitué une ambiance favorable à mes recherches. Un grand merci à Christophe, Gregory, Guillaume, Sébastien, et tous les gens du foyer de l'ENS.

Et puis je n'oublie pas mes parents, qui m'ont aidé tout au long de mes études. Je les embrasse affectueusement.

Enfin, Julie, ma compagne de tous les jours, a permis, par son soutien moral, son support et sa patience, de mener à bien mon travail.

A tous, merci mille fois ; je leur dois beaucoup et je ne l'oublierai pas.



# Table des matières

<b>1 Automates Cellulaires, Systèmes Dynamiques</b>	<b>11</b>
1.1 Automates Cellulaires	11
1.1.1 Généralités	12
1.1.1.1 Topologie de Cantor	14
1.1.2 Classification des Automates Cellulaires	15
1.1.2.1 Classes de Kůrka	15
Équicontinuité	15
Sensibilité aux conditions initiales	18
Expansivité	19
1.1.2.2 Automates Cellulaires Transitifs et Fortement	
Transitifs	19
Transitivité	20
Forte transitivité	20
1.1.2.3 Exemples	21
1.1.3 Agencement des Classes	23
1.2 Topologie de Besicovitch	25
1.2.1 Définition	25
1.2.2 Propriétés des Automates Cellulaires	26
1.2.2.1 Points Fixes et Périodiques	27
1.2.2.2 Séquences de Toeplitz	34
1.2.2.3 Habitation des classes	37
<b>2 Chaos dans les AC : approche algorithmique</b>	<b>41</b>
2.1 Complexité Algorithmique	42
2.1.1 Lemmes techniques sur la complexité algorithmique	44
2.2 Distance Algorithmique sur les configurations bi-infinies	47
2.2.1 Propriétés Topologiques de l'Espace Quotient	47
2.2.2 Automates Cellulaires et Continuité de l'Information	52
<b>3 Régularité et récursivité des Pavages</b>	<b>57</b>
3.1 Généralités sur les pavages	57
3.1.1 Tuiles de Wang et autres modèles de pavage	58
3.1.1.1 Tuiles colorées, ou tuiles de Wang	58
3.1.1.2 Tuiles fléchées	58
3.1.1.3 Formes dont les sommets sont à coordonnées rationnelles	60
3.1.1.4 Contraintes locales	61
3.1.2 Résultats sur les pavages	61

3.1.2.1	Procédé d'extraction diagonale, lemme de König	61
3.1.2.2	Rapports entre les différentes définitions . . . . .	62
3.1.2.3	Problème de pavabilité . . . . .	65
3.1.3	Périodicité et apériodicité . . . . .	66
3.1.3.1	Pavages périodiques . . . . .	66
3.1.3.2	Pavages quasipériodiques . . . . .	68
3.2	Quantification de la quasipériodicité . . . . .	74
3.2.1	Étude de la fonction de quasipériodicité . . . . .	74
3.2.2	Pavages dont la quasipériodicité est non récursivement bornée . . . . .	78
3.3	Variations sur le théorème de Rice . . . . .	79
3.3.1	Théorème de Rice pour les programmes . . . . .	79
3.3.2	Théorème de Rice pour les pavages . . . . .	80
3.3.3	Théorème de Rice pour les automates cellulaires . . . . .	83
	<b>Bibliographie</b>	<b>89</b>
	<b>Index</b>	<b>93</b>

# Introduction

Dans ce travail de thèse, on étudie la complexité d'objets discrets : les pavages à l'aide de tuiles colorées et les automates cellulaires sur une grille. Ces deux notions sont très proches dans leur structure et les pavages sont parfois considérés comme une version statique des automates cellulaires. On espère ainsi pouvoir construire des démonstrations en transposant les preuves d'un monde à un autre. On utilise des arguments directement liés à la structure de ces entités, mais aussi des arguments topologiques et des arguments algorithmiques.

L'aspect qui nous intéresse dans les pavages a été révélé lors de la découverte d'ensembles de tuiles pouvant paver le plan, mais n'admettant aucun pavage périodique. En effet, on conjecturait, dans les années 60, que la pavabilité du plan était décidable. Si cela avait été le cas, tout ensemble de tuiles pavant le plan le paverait aussi périodiquement. On a découvert par la suite qu'il existe des ensembles de tuiles "intermédiaires", appelés ensembles de tuiles apériodiques, qui pavent le plan, mais jamais périodiquement. Ce fut Berger qui construisit le premier exemple d'ensemble de tuiles apériodique dans [Ber66] grâce auquel il parvint à montrer que le problème de la pavabilité du plan (*domino problem*) était indécidable. La profondeur de ce résultat fut concrétisée grâce à la fermeture, par réduction à ce problème, d'un champ de questions ouvertes en logique (voir [BGG96]). De plus, cet ensemble de tuiles apériodique, ainsi qu'un autre, plus simple et plus facile à comprendre introduit par Robinson dans [Rob71], donna naissance à une autre notion fondamentale : celle de quasipériodicité. Dans les pavages quasipériodiques, les motifs présents se répètent de façon régulière dans le plan, mais leur fréquence diminue quand leur taille augmente. L'intérêt de cette notion ne se limite pas au simple domaine de l'informatique ou même à celui des mathématiques, mais s'étend à d'autres domaines tels que la cristallographie, en fournissant un modèle pour les quasicristaux. En effet, on ne comprenait pas comment des cristaux pouvaient avoir une structure symétrique d'ordre cinq. À la suite de ces travaux, Penrose apporta une réponse en construisant son célèbre pavage. Il faut cependant noter que, les tuiles composant ce pavage n'étant pas à coordonnées rationnelles, le modèle de pavage est plus général que celui des tuiles colorées. Par la suite, Bruno Durand montra dans [Dur99] que tout pavage pavant le plan admet un pavage quasipériodique, résultat similaire à ce que l'on conjecturait à propos des pavages périodiques. La notion de quasipériodicité mieux comprise, on a voulu décrire le degré de complexité de ces pavages quasipériodiques. Dans ce but, la définition de fonction de quasipériodicité fut introduite. Cette fonction quantifie la fréquence d'apparition des motifs carrés de taille fixée, de telle sorte que plus la croissance de cette fonction est forte, plus la fréquence des motifs diminue rapidement en fonction de leur taille. Cependant, on ne connaît encore que très peu d'ensembles de



tuiles aperiodiques. La fonction de quasiperiodicite d'un pavage periodique est  $x \mapsto x + c$ , ou  $c$  est une constante; celles des pavages de Robinson et Berger sont lineaires. Notre travail de recherche consiste en l'exploration des classes de fonctions capturables par un ensemble de tuiles, c'est-a-dire la recherche des fonctions que l'on peut obtenir comme fonction de quasiperiodicite d'un pavage. Il est clair que l'on ne peut se contenter de ne regarder que les pavages, sans tenir compte des ensembles de tuiles qui les ont produits. En effet, il est possible de concevoir un ensemble de tuiles colorées élémentaire qui permet de produire n'importe quelle configuration, de la plus simple à la plus compliquée. N'importe quelle fonction de quasiperiodicite est donc atteignable. Ainsi, il est naturel d'étudier quelle est la fonction de quasiperiodicite "minimale" parmi celles des pavages quasiperiodiques, produits par un ensemble de tuiles. Notre étude nous permet de construire, pour toute fonction récursive croissante raisonnable, un ensemble de tuiles dont tous les pavages sont quasiperiodiques, et admettent cette fonction comme fonction de quasiperiodicite. De plus, on construit aussi un ensemble de tuiles dont tous les pavages sont quasiperiodiques, et dont la fonction de quasiperiodicite n'est bornée par aucune fonction récursive. Il n'est donc pas nécessaire d'introduire la minimalité pour résoudre ce problème.

Depuis le *domino problem*, de nombreux autres problèmes de pavages ont été prouvés indécidables. On cherche donc à trouver un résultat plus général qui englobe l'indécidabilité d'une large classe de problèmes concernant les pavages. On espère ainsi démontrer un résultat analogue au théorème de Rice. Ce dernier stipule que toute propriété sur les fonctions récursives est soit indécidable soit triviale. Néanmoins, nous ne parvenons pas à obtenir cet analogue exactement, mais établissons une variante plus faible. Nous utilisons une notion équivalente à celle des ensembles de tuiles colorées, appelée contrainte locale. Cette dernière nous permet d'éliminer les ambiguïtés liées à la comparaison de pavages issus d'ensembles de tuiles différents. La définition de contrainte locale est proche de la notion d'automates cellulaires. On débute ainsi l'analyse de la complexité structurelle de ce modèle de calcul, en montrant, pour les automates cellulaires, un résultat similaire à celui obtenu dans le monde des pavages. Nous poursuivons l'étude de la complexité structurelle des automates cellulaires au travers de l'exploration de considérations topologiques.

La façon dont on perçoit les automates cellulaires comme systèmes dynamiques est très liée à la topologie qu'on utilise. Hedlund a montré qu'une fonction est réalisée par un automate cellulaire si et seulement si elle est continue pour la topologie de Cantor, et invariante par décalage<sup>1</sup>. Ceci donne donc une définition topologique très simple des automates cellulaires. Une mise en valeur de cette approche est due à Čulik *et al.* dans [ČPY89], où ils démontrent un résultat sur les ensembles limites en utilisant le théorème de Baire; aucune autre preuve de ce résultat n'existe à l'heure actuelle. Un bon moyen d'exploiter les apports de la topologie est d'étudier les automates cellulaires comme des systèmes dynamiques. Le cadre classique utilisé est celui de la topologie de Cantor. C'est dans ce cadre que les premières études sur les comportements asymptotiques des automates cellulaires ont été menées. On a tout de suite mis en évidence une grande variété de comportements différents. Certains automates présentaient des évolutions simples et prévisibles, d'autres semblaient

---

<sup>1</sup>L'image d'une configuration décalée d'une case vers la droite est le décalage d'une case vers la droite de l'image de cette configuration

très chaotiques et furent même qualifiés d'aléatoires [Wol86]. C'est alors qu'apparurent les premières tentatives de définition formelle d'automate cellulaire chaotique, en utilisant les définitions classiques des dynamiciens, comme celle d'équicontinuité, ou de sensibilité aux conditions initiales (voir par exemple [Dev89]). Ces définitions ont plusieurs défauts qui vont à l'encontre de l'intuition. En effet, des automates cellulaires très simples, comme par exemple le décalage, sont très chaotiques. Afin de corriger ces défauts, une autre topologie, dite "topologie de Besicovitch", fut introduite dans [CFMM97]; elle permettait notamment de diminuer le poids des cases centrales, et ainsi, de ne pas donner d'importance à une modification minimale d'une configuration qui apparaîtrait en son centre. Cet article, puis d'autres travaux ultérieurs, ont permis de statuer que cet espace convient à l'étude des systèmes dynamiques. Pour débiter notre travail de recherche, nous nous intéressons aux éléments de stabilité des automates cellulaires en étudiant leurs ensembles de points fixes et de points périodiques. On montre que, dans cette topologie, tout automate a soit un point fixe, soit un nombre non dénombrable de points périodiques. Ceci contribue, en partie, à expliquer certains de nos autres résultats, qui montrent qu'il n'existe pas d'automates ayant un degré très élevé de chaotité. Ensuite, on cherche à trouver un sous-ensemble de l'espace qui soit à la fois dense et stable sous l'action de tout automate. On pourra ainsi restreindre nos études à cet ensemble, et grâce à ces propriétés, étendre le résultat à l'espace tout entier. On construit un ensemble des configurations qui convient : les configurations de Toeplitz. On montre que cet ensemble joue un rôle particulier vis-à-vis de la surjectivité et de l'injectivité, qui est fortement similaire à celui des configurations spatialement périodiques dans le cadre de la topologie de Cantor. La dernière partie du chapitre s'inscrit dans le cadre des classes de Kůrka. Ces dernières furent introduites dans [Kůr97] afin de classifier le degré d'équicontinuité des évolutions des automates cellulaires. On peut raffiner ce découpage en introduisant des propriétés plus spécifiques, comme la transitivité, la forte transitivité, ou l'ergodicité. Dans le cadre de la nouvelle topologie, un exemple d'automate cellulaire sensible aux conditions initiales est donné dans [BFK99]. En revanche, il n'y avait aucun exemple d'automate plus chaotique. Par exemple, dans [Man98], le problème de prouver l'existence d'un automate cellulaire transitif dans l'espace de Besicovitch est décrit comme "challenging open problem". À l'aide de la complexité algorithmique, on montre qu'il n'existe pas de tels automates.

Cette première analyse nous a permis de comprendre que si l'on étudie les automates cellulaires sur une topologie plus adaptée qui met en valeur, non pas la présence d'un défaut à un certain endroit, mais la fréquence avec laquelle une telle erreur est produite, on peut alors montrer que la chaotité des automates cellulaires est plus faible : si l'on fait une erreur ponctuelle, les erreurs engendrées par ce défaut initial ont toujours une fréquence nulle. Par la suite, on a voulu montrer que ce fait est bien plus profond. En effet, on prouve que la capacité des automates cellulaires à transformer le contenu informationnel est inexistante, et par conséquent, que toute la complexité algorithmique que l'on peut éventuellement observer dans leurs diagrammes espace-temps est contenue exclusivement dans la configuration initiale. Ainsi, pour mieux tenir compte de l'information contenue dans les configurations des automates cellulaires, on introduit une seconde topologie, définie à l'aide de la complexité algorithmique, encore appelée complexité de Kolmogorov. Cette dernière s'est souvent révélée très fructueuse pour donner des idées nouvelles ou conduire à de nouvelles ap-

plications. Nous avons défini une distance qui mesure la facilité à calculer un élément à l'aide de l'autre, et vice versa. Plus spécifiquement, une séquence infinie sera à faible distance d'une autre si les tailles du plus petit programme nécessaire au calcul d'un préfixe de l'une en fonction du préfixe de l'autre, et du plus petit programme effectuant l'opération inverse, croissent faiblement en fonction de la taille des préfixes. Il s'agit d'une extension au cas infini de différentes distances introduites dans [BGL<sup>+</sup>98]. Dans le cadre de cette nouvelle topologie, nous avons entrepris une étude initiale sur les propriétés de l'espace métrique obtenu, dans le but de vérifier s'il est adapté à l'étude de systèmes dynamiques. Ensuite, on essaye de caractériser les automates cellulaires continus, puis ceux qui ont un comportement chaotique. Nos résultats confirment notre intuition initiale : aucun automate cellulaire n'est en mesure d'ajouter de la complexité au système, ni de la modifier continûment. En particulier, on montre que les automates cellulaires surjectifs, parmi lesquels on trouve les spécimens les plus topologiquement chaotiques, préservent strictement le contenu informationnel.

L'ordre des chapitres de cette thèse ne correspond pas à celui donné dans l'introduction. Plusieurs notions sont à introduire, et commencer par une partie qui n'utilise que les automates cellulaires permet de se familiariser avec cette notion avant d'aborder les parties suivantes. Le second chapitre aborde la complexité algorithmique, et l'utilise dans l'élaboration d'un nouveau cadre d'étude des automates cellulaires. Le dernier traite des pavages, et termine par le parallèle avec les automates cellulaires.

# Chapitre Premier

## Automates Cellulaires, Systèmes Dynamiques

Ce chapitre<sup>1</sup> est dévolu à l'étude de la complexité des automates cellulaires d'un point de vue structurel. Comme déjà évoqué dans l'introduction, et plus en détail dans [CFMM97], il existe plusieurs défaillances aux définitions usuelles du degré de chaoticité des automates cellulaires. On donc étudie leur comportement dans un nouvel espace topologique qui nous semble plus approprié, et qui est fondée sur l'observation du nombre de différences entre deux configurations. Cette étude comporte une partie qui décrit le nouvel espace topologique. Une autre qui cerne quels sont les points périodiques et fixes des automates cellulaires, vecteurs de stabilité. La suivante propose un ensemble de configurations auquel on peut restreindre l'étude d'un automate, est garantissant que la véracité des résultats obtenus peut se transmettre à l'espace tout entier. Enfin, la dernière détermine quels sont les automates cellulaires les plus chaotiques dans notre nouvelle topologie.

### 1.1 Automates Cellulaires

Les automates cellulaires sont un modèle de calcul formel introduit par J. von Neumann pour modéliser l'accroissement cellulaire et l'auto-duplication. Par la suite, ils ont été appliqués avec succès à d'autres disciplines scientifiques, comme les mathématiques, l'informatique, la physique, la chimie ou la géologie. Leur importance provient du fait qu'ils modélisent, à l'aide d'une définition simple, une évolution massivement parallèle que l'on retrouve dans de nombreux cas pratiques : les relations inter-cellulaires en biologie, les interactions inter-moléculaires en physique, etc. La diversité des évolutions observées et leurs comportements complexes et imprévisibles ont posé de nombreuses questions, dont un certain nombre a été traité grâce à la théorie des systèmes dynamiques.

---

<sup>1</sup>ce chapitre fera l'objet de la publication [BCF]

### 1.1.1 Généralités

Un automate cellulaire est un ensemble de cellules contenant chacune un même automate. Chacun possède un état qui évolue à chaque étape du calcul : ce dernier change en fonction de l'état des automates se trouvant dans son voisinage. On peut définir un automate cellulaire sur un graphe en général, et en particulier sur un axe discret, infini ( $\mathbb{N}$ ) ou bi-infini ( $\mathbb{Z}$ ), une grille ( $\mathbb{Z}^2$ ), une grille 3D ( $\mathbb{Z}^3$ ), ou un espace de plus grande dimension.

**Définition 1 (configuration).** *On appelle configuration l'ensemble des états dans lesquels se trouvent les automates qui constituent l'automate cellulaire. Si  $Q$  est l'ensemble des états de l'automate, l'ensemble des configurations d'un automate sur  $\mathbb{Z}^d$  est  $Q^{\mathbb{Z}^d}$ .*

Dans la suite, nous noterons  $X$  l'ensemble des configurations.

**Définition 2 (automates cellulaires sur  $\mathbb{Z}^d$ ).** *Un automate cellulaire  $\mathcal{A}$  sur  $\mathbb{Z}^d$  est un triplet  $(V, Q, \lambda)$ .*

*L'ensemble  $V \subset \mathbb{Z}^d$  est fini et est appelé voisinage de l'automate cellulaire. L'ensemble  $Q$  est fini et est l'ensemble des états de l'automate. L'entier  $d$  est la dimension de l'automate.*

*La fonction  $\lambda : Q^{|V|} \rightarrow Q$  est appelée règle locale de l'automate. Elle induit une règle globale  $f_{\mathcal{A}}$  de la manière suivante : si  $V = (v_1 \dots, v_t)$ ,  $\forall c \in Q^{\mathbb{Z}^d}$  et  $\forall x \in \mathbb{Z}^d$  :*

$$f_{\mathcal{A}}(c)(x) = \lambda(c(x + v_1), \dots, c(x + v_t)) .$$

Cette définition est illustrée figure 1.1.

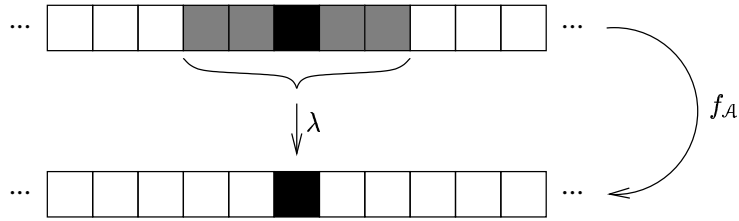


FIG. 1.1 – Fonction globale d'un automate cellulaire, dont le voisinage est  $(-2, -1, 0, 1, 2)$

Dans la suite, quand il n'y a pas de confusion possible, nous assimilons l'automate avec sa règle globale.

**Notation.** *Pour  $a < b$  deux entiers, on notera  $\llbracket a, b \rrbracket$  l'ensemble  $\{a, a+1, \dots, b\}$ .*

Sans perte de généralité, on peut supposer que, pour un automate cellulaire de dimension  $d$ , le voisinage est de la forme  $\llbracket -r, r \rrbracket^d$ . En effet, il suffit de prendre un entier  $r$  tel que  $\llbracket -r, r \rrbracket^d$  contient le voisinage. Dans ce cas, on appelle  $r$  le rayon de l'automate. On a donc  $|V| = (2r + 1)^d$ .

Si  $f$  est un automate cellulaire de rayon  $r$  et de règle locale  $\lambda$ , et  $w$  un mot de taille supérieure à  $2r + 1$ , on notera  $f(w)$  le mot de taille  $|w| - 2r$  défini ainsi, pour tout entier  $i$  entre 1 et  $|w| - 2r$ ,

$$f(w)_i = \lambda(w_i, w_{i+1}, \dots, w_{i+2r})$$

Nous allons utiliser diverses notations afin de manipuler plus simplement les configurations.

**Notation.** Pour  $x \in Q^{\mathbb{Z}}$ , on notera  $x_i = x(i)$  la  $i^{\text{ème}}$  coordonnée de  $x$ . On notera  $x_{a:b}$  le mot  $x_a x_{a+1} \dots x_b$  si  $b \geq a$ . On notera  $\mathcal{M}_x(n)$ , le motif central de taille  $2n + 1$  d'une configuration  $x$ , c'est-à-dire  $x_{-n:n}$ . Dans le cas d'un mot  $u$ , on notera sa taille  $|u|$ ,  $u_1$  sa première lettre,  $u_i$  sa  $i^{\text{ème}}$  lettre. Enfin, si  $\{0, 1\} \subset X$ , on note  $\underline{0}$  la configuration qui ne contient que des 0 et  $\underline{1}$  celle qui ne contient que des 1.

Certaines configurations jouent un rôle particulier dans l'étude des automates cellulaires : les configurations finies et les configurations spatialement périodiques.

**Définition 3 (configuration finie).** Soit  $Q$  un ensemble d'états et 0 un état particulier fixé de  $Q$ . Une configuration  $c$  de  $Q^{\mathbb{Z}^d}$  est finie si elle a un nombre fini de cellules dont l'état est différent de 0.

**Définition 4 (configuration spatialement périodique).** Soit  $Q$  un ensemble d'états. Une configuration  $c$  de  $Q^{\mathbb{Z}^d}$  est spatialement périodique s'il existe un entier  $n$  tel que pour tout  $x \in \mathbb{Z}^d$  et tout  $i$  entre 1 et  $d$ ,  $c(x) = c(x + ne_i)$ , où  $e_i$  est  $i^{\text{ème}}$  vecteur de la base canonique de  $\mathbb{Z}^n$ .

Remarquons que si  $n$  est différent pour chaque dimension, le ppcm de ces  $n$  convient pour vérifier cette définition.

Le fait que les configurations spatialement périodiques et finies soient finiment et récursivement représentables est la raison qui fait intervenir ces définitions dans certains problèmes de décisions concernant les automates cellulaires.

Nous allons étudier les automates cellulaires du point de vue des systèmes dynamiques. L'espace des phases, c'est-à-dire l'espace dans lequel évoluent les systèmes observés, sera donc l'ensemble des configurations, et la fonction successeur sera la règle globale de l'automate cellulaire. Dans ce cadre, on appellera classiquement *point*, un membre de l'espace des phases, c'est-à-dire une configuration.

Trois notions classiques jouent un rôle clef dans l'étude des systèmes dynamiques.

**Définition 5 (point fixe).** On appelle point fixe d'un automate cellulaire  $f$  une configuration  $x$  telle que  $f(x) = x$ .

**Définition 6 (point périodique).** On appelle point périodique d'un automate cellulaire  $f$  une configuration  $x$  telle qu'il existe un entier  $p$  tel que  $f^p(x) = x$  ( $f^p$  est la composition de  $p$  fois la fonction  $f$ ). Le plus petit entier  $p$  vérifiant la propriété précédente est la période du point périodique  $x$ .

**Définition 7 (orbite).** Soit  $x$  une configuration et  $f$  un automate cellulaire. On appelle orbite de  $x$  l'ensemble de points  $\mathcal{O}_f(x) = \{f^i(x), i \in \mathbb{N}\}$  (par convention  $f^0(x) = x$ ).

Dans cette thèse, nous étudions plus en détail le cas de  $\mathbb{Z}$ , quoique la plupart de nos résultats soient aussi vrai dans les dimensions supérieures.

Étant donnée une configuration initiale  $x$ , on appelle *diagramme espace-temps* l'empilement des configurations  $x, f^2(x), \dots, f^n(x), \dots$  (voir Fig. 1.2, Fig. 1.3 et Fig. 1.4). Le *diagramme espace-temps* est une représentation gra-

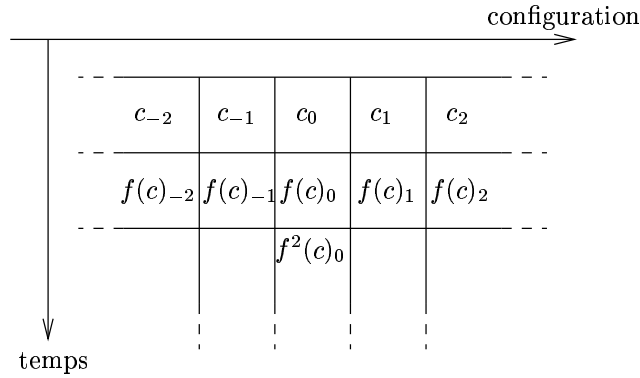


FIG. 1.2 – Diagramme espace-temps

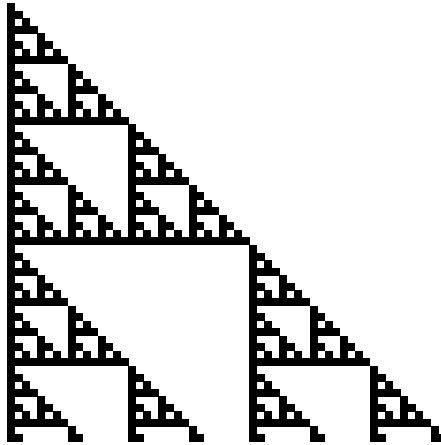


FIG. 1.3 – Diagramme espace-temps de l'automate cellulaire dont chaque cellule fait la somme modulo 2 de son état et de celui de son voisin de gauche

phique des orbites très utile et très employée au niveau des simulations par ordinateur.

#### 1.1.1.1 Topologie de Cantor

L'étude des automates cellulaires, vus comme des systèmes dynamiques, nécessite la mise en place d'une topologie sur l'ensemble des configurations. La topologie la plus couramment utilisée est la topologie de Cantor, c'est-à-dire la topologie produit (plus précisément le produit sur  $\mathbb{Z}$  de la topologie discrète sur  $Q$ ) sur l'espace des configurations.

Si  $w$  est un mot de  $Q^*$  de longueur  $2p + 1$ , on appelle *cylindre* centré sur  $w$ , noté  $[w]$  l'ensemble de configurations suivant :

$$[w] = \{x, \mathcal{M}_x(p) = w\}$$

L'ensemble des cylindres constitue une base d'ouverts de cette topologie.

Il s'agit d'une topologie métrisable. La distance la plus couramment utilisée est la suivante :

$$d(x, y) = 2^{\min\{|i|, x_i \neq y_i, i \in \mathbb{Z}\}} .$$

Plus simplement, deux configurations  $x$  et  $y$  sont à distance supérieure à  $\varepsilon$  l'une de l'autre si  $\mathcal{M}_x(\lceil -\log \varepsilon \rceil) \neq \mathcal{M}_y(\lceil -\log \varepsilon \rceil)$ .

### 1.1.2 Classification des Automates Cellulaires

Lorsque l'on regarde les diagrammes espace-temps de différents automates cellulaires, certains nous paraissent plus "complexes", ou ayant un comportement plus désordonné que d'autres (par exemple, le diagramme Fig. 1.3 semble simple, celui de Fig. 1.4 semble plus imprévisible). C'est à la suite de ce constat que sont apparues les premières études du chaos des automates cellulaires, et les premières définitions tendant à formaliser cette notion.

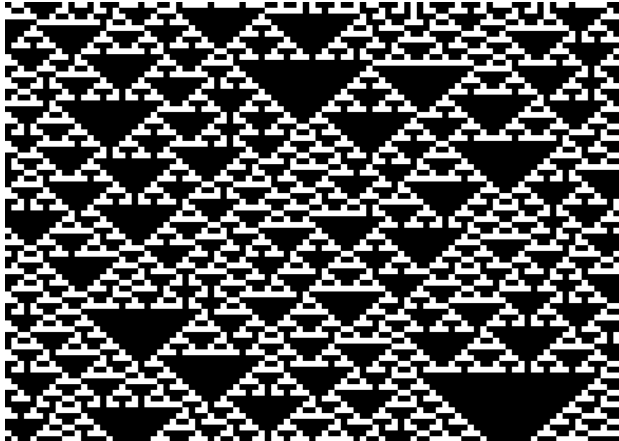


FIG. 1.4 – Diagramme espace-temps d'un automate au comportement plus chaotique

#### 1.1.2.1 Classes de K urka

#####  quicontinuit 

Afin de d crire la stabilit  de l'automate cellulaire, on utilise la notion d' quicontinuit . En particulier, les points fixes et p riodiques sont des points d' quicontinuit , bien qu'il y ait des points d' quicontinuit  qui ne sont ni p riodiques ni fixes.



**Définition 8 (point d'équicontinuité).** Soit  $f$  un automate cellulaire et  $x$  une configuration. L'automate cellulaire est équicontinu en  $x$  si pour tout réel  $\varepsilon > 0$ , il existe un réel  $\delta > 0$  tel que pour tout  $y$  à distance moindre que  $\delta$  de  $x$  et tout entier  $n$ ,  $d(f^n(x), f^n(y)) < \varepsilon$  (Fig. 1.5).

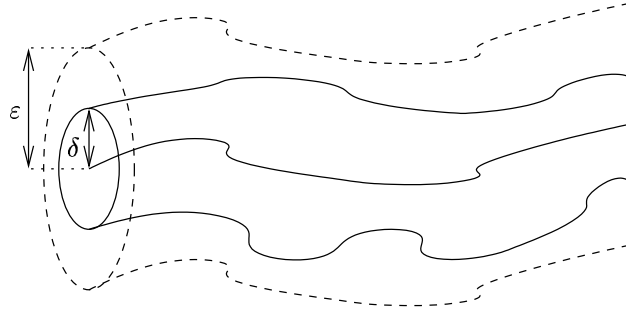


FIG. 1.5 – Équicontinuité

Ainsi, un automate cellulaire est équicontinu en un point  $x$  si les orbites des points proches de  $x$  restent dans un tube autour de l'orbite de  $x$ .

Dans le cadre des automates cellulaires de dimension 1, la notion d'équicontinuité est mieux comprise grâce à la notion de *mot bloquant*. Un mot est *bloquant* si aucune information ne le traverse, ou encore si les évolutions des parties de la configuration situées à droite et à gauche du mot bloquant ne dépendent pas l'une de l'autre. Plus formellement,

**Définition 9 (mot bloquant).** Soit  $f$  un automate cellulaire de dimension 1. Un mot  $w$  est bloquant pour  $f$  s'il existe un entier  $n$  tel que, quelle que soit la configuration  $x$  centrée sur  $w$ , la suite des mots  $\mathcal{M}_{f^i(x)}(n)$ ,  $i \in \mathbb{N}$ , ne dépend que de  $w$  (Fig. 1.6).

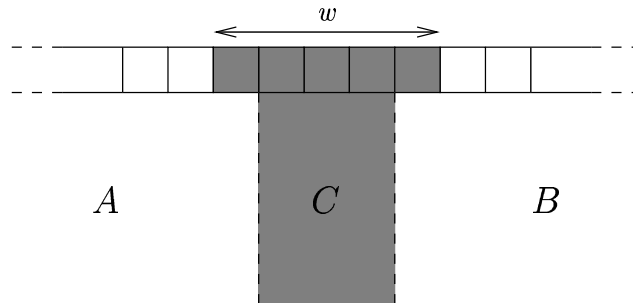


FIG. 1.6 – Diagramme espace-temps d'un automate sur une configuration centrée sur un mot bloquant  $w$  : la région C est fixée par  $w$ , et la région A [resp. B] ne dépend que de la partie de la configuration située à gauche [resp. droite] de  $w$ .

Les propositions suivantes soulignent le lien entre mot bloquant et équicontinuité.

**Proposition 1.** *Si  $x$  est un point d'équicontinuité, il contient un mot bloquant en son centre.*

*Preuve.* Soit  $f$  un automate cellulaire de dimension 1, et  $x$  une configuration qui est un point d'équicontinuité. Alors, pour tout  $\varepsilon > 0$ , il existe un  $\delta > 0$  tel que pour toute configuration  $y$  à distance moindre que  $\delta$  de  $x$ , on a, pour tout entier  $n$ ,  $d(f^n(x), f^n(y)) < \varepsilon$ . Dans la topologie de Cantor, cela signifie que les motifs centraux des configurations  $f^n(x)$  et  $f^n(y)$  de rayon  $\lceil -\log \varepsilon \rceil$  sont identiques pour tout entier  $n$ . On en déduit donc que le motif  $\mathcal{M}_x(\lceil -\log \delta \rceil)$  est un mot bloquant, et que, dans ce cas, la largeur de la colonne invariante est  $2 \lceil -\log \varepsilon \rceil + 1$ .  $\square$

**Proposition 2.** *Si un automate cellulaire  $f$  admet un mot bloquant, alors il a une infinité de points d'équicontinuité.*

*Preuve.* Soit  $f$  un automate cellulaire admettant  $w$  comme mot bloquant. Toute configuration contenant  $w$  une infinité de fois à gauche, et une infinité de fois à droite (pour toute position  $i$ , je peux trouver  $w$  à droite et à gauche de  $i$  dans la configuration) est un point d'équicontinuité. Soit  $x$  une telle configuration. Pour tout  $\varepsilon > 0$ , soit  $\delta > \varepsilon 2^{-|w|}$  tel que le motif  $m$  de taille  $\mathcal{M}_x(\lceil -\log \delta \rceil)$  commence et termine par  $w$ . Dans ce cas, comme  $w$  est un mot bloquant, la suite des mots  $\mathcal{M}_y(\lceil -\log \varepsilon \rceil)$  est la même pour tous les  $y$  de  $[m]$ . Par conséquent,  $\delta$  convient pour vérifier l'hypothèse d'équicontinuité. Le point  $x$  est donc d'équicontinuité.  $\square$

Kůrka a introduit différentes classes afin de déterminer le degré chaotité d'un automate cellulaire. Elles partitionnent l'ensemble des automates cellulaires en quatre classes numérotées de I à IV, de la moins à la plus chaotique :

**classe I :** automates cellulaires équicontinus ;

**classe II :** automates cellulaires admettant des points d'équicontinuités, mais non équicontinus ;

**classe III :** automates cellulaires sensibles aux conditions initiales ;

**classe IV :** automates cellulaires expansifs.

Dans la suite, on reprend chacune d'entre elles, afin d'en donner la définition formelle, et une vision plus intuitive et imagée.

**Définition 10 (automate cellulaire équicontinu, classe I).**

*Un automate cellulaire est équicontinu si pour tout réel  $\varepsilon > 0$ , il existe un réel  $\delta > 0$  tel que pour tous  $x$  et  $y$  tels que  $d(x, y) < \delta$ , et pour tout entier  $n$ ,  $d(f^n(x), f^n(y)) < \varepsilon$ . On dit aussi que cet automate est de classe I.*

La compacité de l'espace nous permet de prouver la proposition suivante :

**Proposition 3.** *Un automate cellulaire est équicontinu si et seulement s'il est équicontinu en tout point.*

*Preuve.* Un sens est évident : si un automate cellulaire est équicontinu, il l'est en tout point.

Réciproquement, soit  $f$ , un automate cellulaire équicontinu en tout point. Montrons qu'il est équicontinu. Soit  $\varepsilon > 0$ . À toute configuration  $x \in X$ , on

associe un réel  $\delta_x > 0$  tel que pour tout  $y$  vérifiant  $d(x, y) < \delta_x$ , et pour tout entier  $n$ ,  $d(f^n(x), f^n(y)) < \frac{\varepsilon}{2}$ . Soit  $\mathcal{R}$ , l'ensemble

$$\mathcal{R} = \{\text{BO}(x, \frac{\delta_x}{2}), x \in X\}$$

où  $\text{BO}(x, r)$  est la boule ouverte de centre  $x$  et de rayon  $r$ . L'ensemble  $\mathcal{R}$  est un recouvrement ouvert de  $X$ ; on en extrait par compacité un sous-recouvrement fini :

$$\exists A \subset X, |A| < \infty, \bigcup_{x \in A} \text{BO}(x, \frac{\delta_x}{2}) = X$$

Soit  $\delta = \min_{x \in A} \frac{\delta_x}{2}$ , qui existe car  $A$  est fini. Montrons que ce  $\delta$  convient pour vérifier la condition d'équicontinuité. Pour tout  $x, y$  tel que  $d(x, y) < \delta$ , comme  $\mathcal{R}$  est un recouvrement ouvert de  $X$ , il existe  $x'$  et  $y'$  dans  $A$  tel que  $d(x, x') \leq \frac{\delta_{x'}}{2}$  et  $d(y, y') \leq \frac{\delta_{y'}}{2}$ . On a donc,

$$d(x, x') < \frac{\delta_{x'}}{2} < \delta_{x'} \quad , \quad (1.1)$$

et comme  $d(x, y) < \delta \leq \frac{\delta_{x'}}{2}$ ,

$$d(x', y) < d(x', x) + d(x, y) < \delta_{x'} \quad . \quad (1.2)$$

Ainsi, par définition de  $\delta_{x'}$ , grâce aux équations (1.1) et (1.2), on obtient que pour tout  $n$  :

$$d(f^n(x'), f^n(x)) < \frac{\varepsilon}{2} \quad \text{et} \quad d(f^n(x'), f^n(y)) < \frac{\varepsilon}{2}$$

si bien que l'on déduit :

$$d(f^n(x), f^n(y)) < d(f^n(y), f^n(x')) + d(f^n(x'), f^n(x)) < \varepsilon \quad .$$

□

### Définition 11 (automate cellulaire de classe II).

*Un automate cellulaire est de classe II s'il possède des points d'équicontinuité, sans être de classe I.*

### Sensibilité aux conditions initiales

#### Définition 12 (automate cellulaire sensible, classe III).

*Un automate cellulaire est sensible aux conditions initiales s'il existe un réel  $\varepsilon > 0$  tel que pour tout point  $x$ , et tout voisinage topologique  $V$  de  $x$ , il existe  $y \in V$  et  $n \in \mathbb{N}$  tel que  $d(f^n(x), f^n(y)) > \varepsilon$  (Fig. 1.7). Le réel  $\varepsilon$  est appelé constante de sensibilité. On dit aussi que cet automate est de classe III.*

Ainsi, un automate cellulaire est sensible aux conditions initiales si une perturbation du point de départ, aussi faible soit elle, peut modifier complètement le comportement de l'automate.

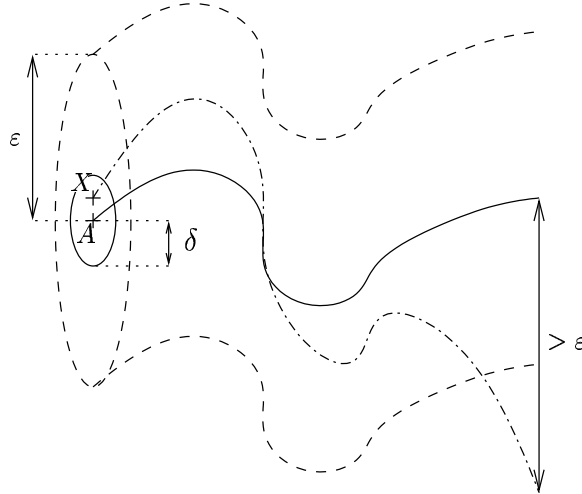


FIG. 1.7 – Un automate cellulaire est sensible aux conditions initiales s'il existe  $\varepsilon$  tel que pour tout point  $A$  et tout  $\delta$ , il existe une configuration  $X$  dont l'orbite sort du tube de rayon  $\varepsilon$  centré sur l'orbite de  $A$

### Expansivité

#### Définition 13 (automate cellulaire expansif, classe IV).

Un automate cellulaire  $f$  est expansif s'il existe un réel  $\varepsilon > 0$  tel que, quelles que soient les configurations  $x$  et  $y$ , il existe un entier  $n$  tel que  $d(f^n(x), f^n(y)) > \varepsilon$  (Fig. 1.8). Le réel  $\varepsilon$  est aussi appelé constante d'expansivité. On dit aussi que cet automate est de classe IV.

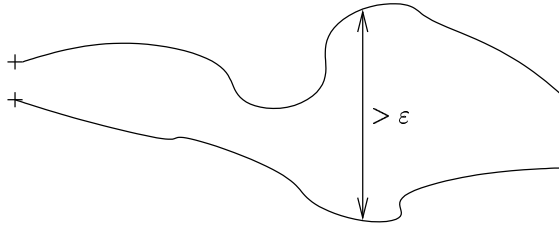


FIG. 1.8 – Expansivité

Un automate cellulaire est expansif si quels que soient les points de départs, deux trajectoires finissent toujours par se séparer d'au moins  $\varepsilon$ .

#### 1.1.2.2 Automates Cellulaires Transitifs et Fortement Transitifs

En plus des quatre classes introduites ci-dessus, deux autres notions ont été créées afin de raffiner la description du comportement des automates cellulaires chaotiques.

**Transitivité**

**Définition 14 (automate cellulaire transitif).** *Un automate cellulaire  $f$  est transitif si pour tous ouverts  $U$  et  $V$ , il existe  $x \in U$  et  $n \in \mathbb{N}$  tel que  $f^n(x) \in V$  (Fig. 1.9).*

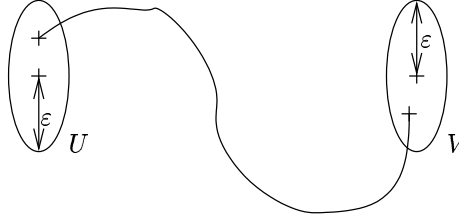


FIG. 1.9 – Transitivité – comme la topologie est métrique, on peut, sans perte de généralité, restreindre la définition en choisissant  $U$  et  $V$  boules ouvertes de même rayon  $\varepsilon$ .

Cette définition peut aussi être énoncée ainsi : pour tout  $\varepsilon > 0$  et pour toutes configurations  $x$  et  $y$ , il existe deux configurations  $x'$  et  $y'$  et un entier  $n$  tel que  $f^n(x') = y'$ .

Ainsi, un automate cellulaire est *transitif* si on peut arriver aussi près que l'on veut d'un point en partant d'un voisinage topologique aussi petit que l'on veut d'un autre.

**Forte transitivité**

**Définition 15 (automate cellulaire fortement transitif).**

*Un automate cellulaire  $f$  est fortement transitif si pour tout ouvert  $U$  et toute configuration  $y$ , il existe  $x \in U$  et  $n \in \mathbb{N}$  tel que  $d(x, x') < \varepsilon$  et  $f^n(x') = y$  (Fig. 1.10).*

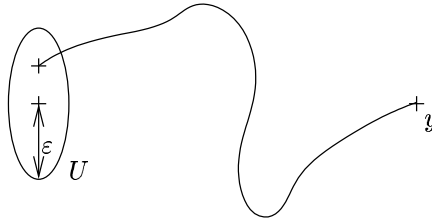


FIG. 1.10 – Forte transitivité

À la différence de la transitivité, ici, on impose à l'orbite de passer exactement par un point, et non dans n'importe quel voisinage topologique.

**Remarque 1.** *Un automate cellulaire fortement transitif ne peut être injectif. En effet, soit  $f$  un automate cellulaire fortement transitif. Par définition des automates cellulaires, l'ensemble  $\{\underline{1}, \underline{0}\}$  est stable par  $f$ . Soit  $U$  un ouvert ne contenant ni  $\underline{1}$  ni  $\underline{0}$ , qui existe dans toute topologie séparée. Il y a donc trois cas possible :*

- $\underline{0} \xrightarrow{f} \underline{1} \xrightarrow{f} \underline{0}$ . Comme, par transitivité, il existe une configuration  $x$  de  $U$  et un entier  $n$  tel que  $f^n(z) = \underline{0}$  et  $f$  n'est pas injective. En effet, si  $m$  est le plus grand entier tel que  $f^m(z) \notin \{\underline{1}, \underline{0}\}$ , alors  $f^{m+1}(z)$  a deux antécédents :  $f^m(z)$  et l'élément de  $\{\underline{1}, \underline{0}\}$  qui n'est pas  $f^{m+1}(z)$  ;
- $\underline{0} \xrightarrow{f} \underline{1} \xrightarrow{f} \underline{1}$ . Dans ce cas,  $f$  n'est pas injective ;
- $\underline{0} \xrightarrow{f} \underline{0}$ . Par transitivité, il existe une configuration  $x$  de  $U$  et un entier  $n$  tel que  $f^n(z) = \underline{0}$  et  $f$  n'est pas injective, par le même argument que dans le premier cas.

### 1.1.2.3 Exemples

**Exemple 1 (décalage : automate cellulaire transitif).** Considérons  $f$ , l'automate cellulaire de rayon 1 à deux états 0 et 1, et de règle locale

$$\lambda(x_{-1}, x_0, x_1) = x_1 .$$

Cet automate cellulaire particulier est appelé de décalage à gauche. Soit  $f'$ , l'automate cellulaire de rayon 1, à deux états 0 et 1 et de règle locale

$$\lambda'(x_{-1}, x_0, x_1) = x_{-1} .$$

Cet automate cellulaire est appelé de décalage à droite. On le note  $\sigma$ .

Les décalages sont transitifs, non fortement transitifs (dans le cas bi-infini).

Les décalages jouent un rôle particulier dans l'étude des systèmes dynamiques, car ils sont impliqués dans la définition topologique des automates cellulaires : un automate cellulaire est une fonction continue dans la topologie de Cantor, invariante par décalage.

### Exemple 2 (automate cellulaire fortement transitif, non expansif).

Considérons  $f$ , l'automate cellulaire de rayon 1, à deux états 0 et 1 et de règle locale  $\lambda(x_{-1}, x_0, x_1) = x_0 \oplus x_1$  où  $\oplus$  est l'addition modulo 2 (Fig. 1.3).

Remarquons que pour toute configuration  $x$ , l'état de la  $i^{\text{ème}}$  cellule de  $f(x)$  ne dépend que des états des cellules situées à droite de la  $i^{\text{ème}}$  de  $x$ , et de son propre état. Ainsi, seul ce qui est situé à droite de la cellule peut influencer son comportement. Par conséquent, pour tout  $\varepsilon > 0$ , si on choisit  $x = \underline{0}$  et  $y$  défini par :

$$\forall i \in \mathbb{Z}, y_i = \begin{cases} 1 & \text{si } i \leq \lceil \log \varepsilon \rceil \\ 0 & \text{sinon} \end{cases} ,$$

alors les motifs  $f^i(x)_{-\lceil \log \varepsilon \rceil : \lceil \log \varepsilon \rceil}$  et  $f^i(y)_{-\lceil \log \varepsilon \rceil : \lceil \log \varepsilon \rceil}$  sont les identiques car  $y$  et  $x$  sont égaux de  $\lceil \log \varepsilon \rceil$  à  $\infty$ . Par conséquent,  $d(f^i(x), f^i(y)) < \varepsilon$ . Il n'existe donc pas de constant d'expansivité, et donc  $f$  n'est pas expansif.

Montrons que  $f$  est fortement transitif. L'illustration de cette preuve est la figure 1.11. Pour cela, montrons par récurrence sur  $q > 0$ , la propriété  $P(q)$  suivante : "soit  $w$  un motif de taille  $q$  donné, que l'on suppose présent dans une configuration  $x$  à la position  $i$  ( $x_{i:i+q-1} = w$ ). Alors :

- i) l'état  $f^{q-1}(x)_i$  ne dépend que de  $w$  ;
- ii) la fonction qui à l'état  $x_{q+i}$  associe l'état  $f^q(x)_i$  est définie et bijective ;
- iii) la fonction qui à l'état  $x_{i-1}$  associe l'état  $f^q(x)_{i-1}$  est définie et bijective."

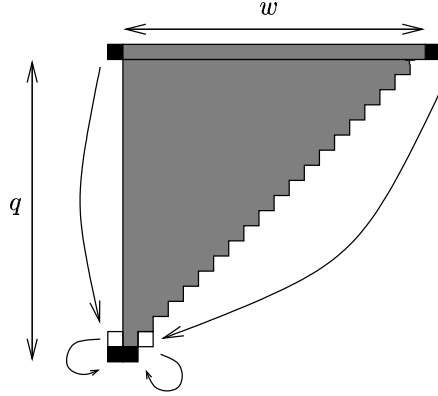


FIG. 1.11 – Illustration de l'étape  $P(n-1) \implies P(n)$  : la zone grisée ne dépend que de  $w$ .

La propriété  $P(1)$  est immédiatement vraie : soit  $w$  un mot constitué d'un seul état et  $x$  telle que  $x_i = w_1$ ,

- i) l'état  $x_i$  ne dépend que de  $w = x_i$  ;
- ii) la fonction qui à l'état  $x_{1+i}$  associe  $f(x)_i = w_1 \oplus x_{1+i}$  est définie et bijective ;
- iii) la fonction qui à l'état  $x_{i-1}$  associe  $f(x)_{i-1} = x_{i-1} \oplus w_1$  est définie et bijective ;

Pour  $q > 1$ , supposons la propriété  $P(q-1)$  vraie. Soit  $w$  un mot de taille  $q$  supposé inclus dans une configuration  $x$  à la position  $i$ .

- i) On a  $f^{q-1}(x)_i = f^{q-2}(x)_i \oplus f^{q-2}(x)_{i+1}$ . D'après le point i) de  $P(q-1)$  appliqué aux mots  $w_{2:|w|}$  et  $w_{1:|w|-1}$ , on en déduit que  $f^{q-2}(x)_i$  et  $f^{q-2}(x)_{i+1}$  ne dépendent que de  $w$ . Par conséquent,  $f^{q-1}(x)_i = f^{q-2}(x)_i \oplus f^{q-2}(x)_{i+1}$  ne dépend que de  $w$  ;
- ii) on a  $f^q(x)_i = f^{q-1}(x)_i \oplus f^{q-1}(x)_{i+1}$ . Grâce au point i) ci-dessus, la fonction qui à  $f^{q-1}(x)_{i+1}$  associe  $f^q(x)_i$  est définie et bijective. D'après le point ii) de  $P(q-1)$  appliqué aux mots  $w_{2:|w|}$ , on sait que la fonction qui à  $x_{|w|+i}$  associe  $f^{q-1}(x)_{i+1}$  est définie et bijective. Par composition, la fonction qui à  $x_{|w|+i}$  associe  $f^q(x)_i$  est définie et bijective ;
- iii) on a  $f^q(x)_{i-1} = f^{q-1}(x)_{i-1} \oplus f^{q-1}(x)_i$ . Grâce au point i) ci-dessus, la fonction qui à  $f^{q-1}(x)_{i-1}$  associe  $f^q(x)_{i-1}$  est définie et bijective. D'après le point iii) de  $P(q-1)$  appliqué aux mots  $w_{1:|w|-1}$ , on sait que la fonction qui à  $x_{i-1}$  associe  $f^{q-1}(x)_{i-1}$  est définie et bijective. Par composition, la fonction qui à  $x_{i-1}$  associe  $f^q(x)_{i-1}$  est définie et bijective.

Soit  $x$  une configuration et  $\varepsilon > 0$ . Soit  $q = \lceil -\log \varepsilon \rceil$  et  $v = x_{-q:q}$ . Soit  $x'$  tel que  $x'_{-q:q}$  est fixé et égal à  $v$ . La fonction qui à  $(x'_{-\infty:-q-1}, x'_{q+1:\infty})$  associe  $f^{2q+1}(x')$  est définie et surjective.

Pour cela, montrons par récurrence sur  $n > 0$  la propriété  $P'(n)$  qui dit : "la fonction qui à  $(x'_{-n-q:-q-1}, x'_{q+1:n+q})$  associe  $f^{2q+1}(x')_{-n-q:n-q-1}$  est définie et bijective." Cette preuve est illustrée par la figure 1.12.

Pour  $n = 1$ , c'est la propriété  $P(2q+1)$  appliquée avec  $w = v$  ;

pour  $n \geq 2$  fixé, supposons la propriété  $P'(n-1)$  vraie, c'est-à-dire que la fonction qui à  $(x'_{-n-q+1:-q-1}, x'_{q+1:n+q-1})$  associe  $f^{2q+1}(x')_{-n-q+1:n-2-q}$  est définie et bijective. Grâce à la propriété  $P(2q+1)$  point ii) appliquée pour  $w$  valant  $x'_{n-q-1:n+q-1}$ , la fonction qui à  $x'_{n+q}$  associe  $f^{2q+1}(x')_{n-q-1}$  est définie et bijective. Grâce à la propriété  $P(2q+1)$  point iii) appliquée avec  $w$  valant  $x'_{-n-q-1:-n+q-1}$ , la fonction qui à  $x'_{-n-q}$  associe  $f^{2q+1}(x')_{-n-q}$  est définie et bijective. Ainsi, la fonction qui à  $(x'_{-n-q:-q-1}, x'_{q+1:n+q})$  associe  $f^{2q+1}(x')_{-n-q:n-q-1}$  est définie et bijective, autrement dit  $P'(n)$  est vraie.

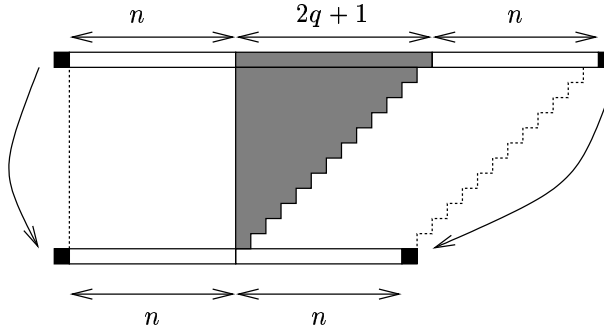


FIG. 1.12 – Illustration de la preuve de  $P'(n-1) \implies P'(n)$

Comme cette propriété est vraie pour tout  $n$ , on déduit que la fonction qui à  $(x'_{-\infty:-q-1}, x'_{q+1:\infty})$  associe  $f^{2q+1}(x')$  est définie et surjective. Les états des positions  $-n-q$  et  $n+q$  de  $f^{2q+1}$  sont donnés par  $P'(n)$  (comme dans ce cas la fonction est bijective, les états de ces positions donnés par  $P'(i)$  pour  $i > n$  sont les mêmes).

Ainsi, pour tout  $y$ , il existe une configuration  $x'$  tel que  $x'_{-q:q} = x_{-q:q}$ , c'est-à-dire  $d(x, x') \leq \varepsilon$  tel que  $f^{2q+1}(x') = y$ . L'automate cellulaire  $f$  est fortement transitif.

**Exemple 3 (automate cellulaire expansif).** Considérons l'automate cellulaire  $f$  de rayon 1, à deux états 0 et 1 et de règle locale  $\lambda(x_{-1}, x_0, x_1) = x_{-1} \oplus x_1$  où  $\oplus$  est l'addition modulo 2.

Cet automate est donné dans la littérature comme exemple classique d'expansif.

### 1.1.3 Agencement des Classes

Il est immédiat que, l'espace des phases étant parfait (c'est-à-dire sans points isolés), un automate expansif est sensible aux conditions initiales; il est aussi clair qu'un automate cellulaire fortement transitif est aussi transitif.

Le résultat suivant est dû à Kůrka [Kůr97], mais on le redémontre ici par soucis de complétude, et avec une preuve plus "visuelle".

**Théorème 1.** *Un automate cellulaire de dimension 1 transitif est sensible aux conditions initiales.*

*Preuve.* Par l'absurde, supposons qu'il existe un automate cellulaire  $f$  transitif non sensible aux conditions initiales. S'il n'est pas sensible aux conditions



initiales, il admet donc des points d'équicontinuité et par conséquent un mot bloquant  $w$ . Considérons le cylindre  $[w0w]$ , où  $0$  est un état quelconque de l'automate. Comme  $w$  est bloquant, les mots de taille  $2|w| + 1$  situés au centre des configurations de  $f^n([w0w])$ ,  $n \in \mathbb{N}$  sont de la forme  $axbxc$ , où  $a$ ,  $b$ ,  $c$  et  $x$  sont des mots, et  $x$  de taille non nulle (Fig. 1.13). Par conséquent, aucune des configurations de  $[w0w]$  ne mène au cylindre  $[1^{|w|}00^{|w|}]$ , où  $1$  et  $0$  sont des états de l'automate. Par conséquent, l'automate n'est pas transitif.  $\square$

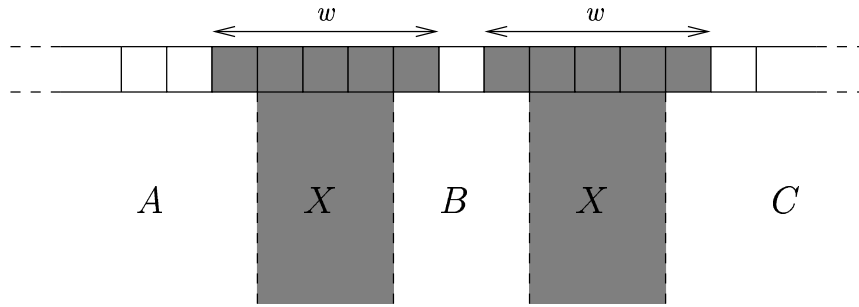


FIG. 1.13 – Les deux zones grisées marquées  $X$  sont identiques

On peut montrer le théorème suivant, qui, en conjonction avec le précédent, prouve que l'expansivité est une propriété très forte.

**Théorème 2.** *Un automate cellulaire expansif est fortement transitif.*

*Preuve.* Soit  $f$  un automate cellulaire expansif, d'ensemble d'états  $Q$  et de rayon  $r$ . Il est donc topologiquement conjugué au décalage  $\sigma$  sur un sous-shift unilatère  $S$ , sous-ensemble fermé et invariant par décalage de  $(A^r)^{\mathbb{N}}$ . De plus,  $S$  est un sous-shift mélangeant de type fini ([Nas95]).

On notera  $L(S)$  l'ensemble des motifs finis des configurations de  $S$ . Un sous-shift de type fini mélangeant vérifie la propriété suivante, appelée *propriété de spécification* : il existe un entier  $n_0 > 0$  tel que pour tous mots  $u$  et  $v$  de  $L(S)$ , et pour tout  $n \geq n_0$ , il existe un mot  $w \in L(S)$  de taille  $n$  tel que  $uwv \in L(S)$ .

On en déduit que pour tout mot  $u \in L(S)$ , toute configuration  $x \in S$  et tout entier  $n \geq n_0$ , il existe un mot  $w \in L(S)$  de taille  $n$  tel que  $uwx \in S$ . En effet, pour tout entier  $i$ , il existe  $w_i \in L(S)$  de taille  $n$  tel que  $uw_i x_{0:i} \in L(S)$ . Soit  $w$  un mot qui apparaît une infinité de fois parmi les  $w_i$ , et  $I$  l'ensemble des  $i$  tels que  $w_i = w$ . Si l'on note, pour  $i \in I$ ,  $c_i = uw_i x_{0:i} 0^\infty$ , comme  $d(uwx, c_i) \leq 2^{-i-|u|}$ , on en déduit que la limite est  $uwx$ . Comme  $S$  est fermé, la suite  $c$  converge dans  $S$ , et donc que  $uwx \in S$ .

Montrons que  $f$  est fortement transitif. Ainsi, soit  $C$  un ouvert. Sans perte de généralité, on peut supposer que  $C = \{ux, x \in S\}$ ,  $u$  étant un mot de  $L(S)$ , car l'ensemble des ouverts de ce type est une base d'ouverts. Pour tout  $x \in S$ , il existe un mot  $w$  tel que  $uwx \in S$  qui est donc un antécédent de  $x$  par  $\sigma^{n+|u|}$  inclus dans cet ouvert. Le décalage  $\sigma$  sur  $S$  est donc fortement transitif, et son conjugué  $f$  aussi.  $\square$

## 1.2 Topologie de Besicovitch

L'étude des automates cellulaires en tant que système dynamique permet l'analyse de comportement asymptotique des automates, des plus simples aux plus chaotiques. Il est cependant difficile de les classer à cause de la très grande diversité de leur comportements. Cette difficulté est d'autant plus importante dans le cas des automates cellulaires chaotiques, car, non seulement, il est difficile de comprendre le comportement d'un automate chaotique quelconque, mais aussi, il n'existe pas de définition "canonique" d'automate chaotique. Notamment, on constate que les définitions précédentes ne sont pas satisfaisantes. En effet, un automate cellulaire très simple comme le décalage est considéré comme chaotique. L'automate additif qui effectue une addition modulo deux est expansif, classe la plus chaotique, alors que son comportement est simplement calculable, en effectuant une exponentiation de polynômes, pour laquelle il existe de nombreuses simplifications dues à des résultats d'algèbre. Pour palier ce problème, on utilise la topologie de Besicovitch, qui permet de diminuer l'importance du motif central lors de l'étude de l'évolution de l'automate, car elle tend à donner à toutes les cases un poids égal en moyennant les effets ([CFMM97]). La topologie utilisée pour l'étude d'un système dynamique est cruciale car son choix peut complètement modifier notre appréciation du comportement du système.

### 1.2.1 Définition

**Définition 16 (pseudo-distance de Besicovitch).**

*On appelle pseudo-distance de Besicovitch, notée  $d_{\mathcal{B}}$  ou simplement  $d$  quand cela ne prête pas à confusion, la fonction suivante :*

$$d_{\mathcal{B}}(x, y) = \limsup_{n \rightarrow \infty} \frac{\Delta(x_{-n:n}, y_{-n:n})}{2n + 1}$$

où  $\Delta(u, v)$  est le nombre de cases où les mots  $u$  et  $v$  diffèrent.

Remarquons que si, pour deux lettres  $a$  et  $b$ , on note :

$$\delta(a, b) = \begin{cases} 0 & \text{si } a = b \\ 1 & \text{sinon} \end{cases}$$

alors pour tous mots  $x$  et  $y$  de taille  $n$ ,

$$\Delta(x, y) = \sum_{i=1}^n \delta(x_i, y_i).$$

Contrairement aux distances, les pseudo-distances ne vérifient que deux axiomes, à savoir le fait que la distance d'un point à lui-même est nulle et l'inégalité triangulaire. Montrons que  $d_{\mathcal{B}}$  est une pseudo-distance.

– Vérifions que pour toute configuration  $x$ ,  $d_{\mathcal{B}}(x, x) = 0$  :

$$\begin{aligned} \forall x, d_{\mathcal{B}}(x, x) &= \limsup_{n \rightarrow \infty} \frac{\Delta(x_{-n:n}, x_{-n:n})}{2n + 1} \\ &= \limsup_{n \rightarrow \infty} \frac{0}{2n + 1} \\ &= 0 \end{aligned}$$

- Vérifions que  $d_{\mathcal{B}}$  vérifie l'inégalité triangulaire, pour toutes configurations  $x, y$  et  $z$  :

$$d_{\mathcal{B}}(x, y) + d_{\mathcal{B}}(y, z) = \limsup_{n \rightarrow \infty} \frac{\Delta(x_{-n:n}, y_{-n:n}) + \Delta(y_{-n:n}, z_{-n:n})}{2n + 1}$$

Or, si  $u, v$  et  $w$  sont trois mots de même taille,  $\Delta(u, w) \leq \Delta(u, v) + \Delta(v, w)$ .

En effet, c'est vrai pour chaque position  $i$  :

- Si  $u_i = v_i = w_i$ , alors  $\delta(u_i, w_i) = 0 = \delta(u_i, v_i) + \delta(v_i, w_i)$  ;
- Si  $u_i = v_i \neq w_i$ , alors  $\delta(u_i, w_i) = 1 = \delta(u_i, v_i) + \delta(v_i, w_i)$  ;
- Si  $u_i \neq v_i = w_i$ , alors  $\delta(u_i, w_i) = 1 = \delta(u_i, v_i) + \delta(v_i, w_i)$  ;
- Si  $u_i = w_i \neq v_i$ , alors  $\delta(u_i, w_i) = 0 < \delta(u_i, v_i) + \delta(v_i, w_i) = 2$ .

d'où, quelles que soient les configurations  $x, y$  et  $z$  :

$$\begin{aligned} d_{\mathcal{B}}(x, y) + d_{\mathcal{B}}(y, z) &= \limsup_{n \rightarrow \infty} \frac{\Delta(x_{-n:n}, y_{-n:n}) + \Delta(y_{-n:n}, z_{-n:n})}{2n + 1} \\ &\geq \limsup_{n \rightarrow \infty} \frac{\Delta(x_{-n:n}, z_{-n:n})}{2n + 1} \\ &\geq d_{\mathcal{B}}(x, z) \end{aligned}$$

En revanche, il existe des configurations distinctes à distance 0 l'une de l'autre, par exemple deux configurations qui ne diffèrent qu'en une seule position. Afin d'obtenir une vraie distance, et ainsi d'avoir un espace métrique dans lequel on peut travailler, nous devons quotienter l'espace des configurations.

Nous noterons  $\dot{=}$  la relation suivante :

$$x \dot{=} y \iff d_{\mathcal{B}}(x, y) = 0 .$$

Le fait que  $d_{\mathcal{B}}$  soit une pseudo-distance suffit à prouver que  $\dot{=}$  est une relation d'équivalence.

L'espace des phases que nous allons considérer est donc l'ensemble des classes d'équivalence, que nous noterons :

$$\dot{X} = \frac{X}{\dot{=}}$$

qui sera notre espace de travail.

La topologie de Besicovitch est issue de la distance de Besicovitch sur l'espace  $\dot{X}$ . Une base d'ouverts de cette topologie est l'ensemble des boules ouvertes de  $\dot{X}$ . Elle est bien adaptée à l'étude des automates cellulaires car elle satisfait un certain nombre de propriétés qui la rendent intéressante, comme par exemple : la perfection, la dimension infinie ou la complétude. Cependant, l'étude est rendue plus difficile car la topologie n'est pas localement compacte.

## 1.2.2 Propriétés des Automates Cellulaires

Une question importante lorsque l'on étudie des fonctions dans un espace formé de classes d'équivalence est la compatibilité de ces fonctions avec la relation d'équivalence sous-jacente. En ce qui concerne les automates cellulaires, on a le résultat suivant :

**Proposition 4 ([CFMM97]).** *Tout automate cellulaire  $f$  est compatible avec  $\dot{=}$ , c'est-à-dire que pour toutes configurations  $x$  et  $y$  de  $X$ ,  $x \dot{=} y \implies f(x) \dot{=} f(y)$ .*

*Preuve.* Soient  $x$  et  $y$ , deux configurations équivalentes, et  $\lambda$  la règle locale de  $f$ . On a, pour tout entier  $n$  :

$$\begin{aligned}
\Delta(f(x)_{-n:n}, f(y)_{-n:n}) &= \sum_{i=-n}^n \delta(f(x)_i, f(y)_i) \\
&= \sum_{i=-n}^n \delta(\lambda(x_{i-r}, \dots, x_{i+r}), \lambda(y_{i-r}, \dots, y_{i+r})) \\
&\leq \sum_{i=-n}^n \sum_{j=-r}^r \delta(x_{i+j}, y_{i+j}) \\
&= \sum_{j=-r}^r \sum_{i=-n}^n \delta(x_{i+j}, y_{i+j}) \\
&= \sum_{j=-r}^r \Delta(x_{-n+j:n+j}, y_{-n+j:n+j}) \\
&\leq (2r+1)\Delta(x_{-n+r:n+r}, y_{-n+r:n+r}) .
\end{aligned}$$

En divisant l'équation précédente par  $2n+1$  et en calculant la limite supérieure quand  $n$  tend vers l'infini, on en déduit que  $d(f(x), f(y)) \leq (2r+1)d(x, y) = 0$ .  $\square$

Ainsi, pour tout automate cellulaire  $f$ , on notera  $\dot{f}$  la fonction qui, à toute classe  $x$ , associe la classe contenant les images des configurations de  $x$ .

Dans le cadre des automates cellulaires, on peut montrer certaines propriétés relatives aux points fixes et périodiques. On peut également souligner le parallélisme entre le rôle joué par les configurations périodiques sur la topologie de Cantor, et les séquences de Toeplitz sur la topologie de Besicovitch. Enfin, on complète l'étude des diverses classes d'automates en démontrant un résultat sur les transitifs.

### 1.2.2.1 Points Fixes et Périodiques

Lors de l'étude des points fixes et périodiques, nous allons être amenés à comparer un point et son image. Par conséquent, on utilise les notations suivantes. Si  $p$  est un entier, et  $u$  un mot de taille au moins  $2p$ , alors  ${}_p|u|_p$  est le mot  $u_{p+1:|u|-p}$ , c'est-à-dire le mot  $u$  privé de ses  $p$  premières et  $p$  dernières lettres.

Dans plusieurs preuves, nous avons besoin des lemmes techniques suivants :

**Lemme 1.** *Pour tout automate cellulaire  $f$  de rayon  $r$  et pour tous mots  $a$  et  $b$  de taille supérieure à  $2r$ , on a :*

$$\Delta({}_r|ab|_r, f(ab)) \leq \Delta({}_r|a|_r, f(a)) + 2r .$$

*Preuve.* Ce résultat provient du fait que l'image de la concaténation de deux mots est la concaténation des deux images séparées par  $2r$  cellules. (Fig. 1.14).

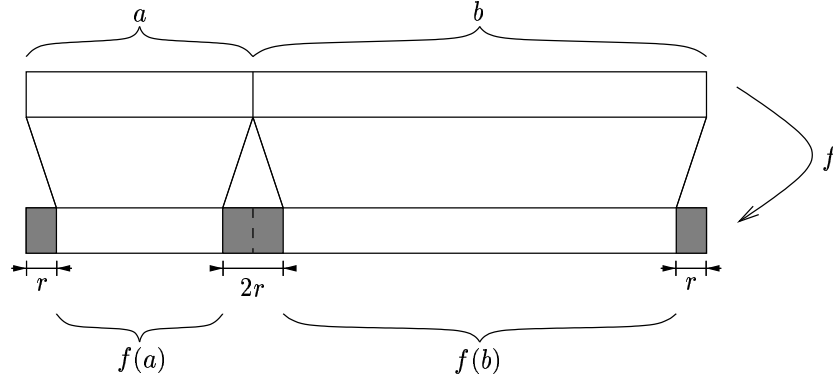


FIG. 1.14 – Illustration du lemme 1

Formellement, soit  $\lambda$  la règle locale de  $f$ , alors :

$$\begin{aligned}
 \Delta(r|ab|_r, f(ab)) &= \sum_{i=r+1}^{|a|-r} \delta(a_i, f(ab)_i) + \sum_{i=|a|-r}^{|a|} \delta(a_i, f(ab)_i) \\
 &\quad + \sum_{i=1}^r \delta(b_i, f(ab)_{|a|+i}) + \sum_{i=r+1}^{|b|-r} \delta(b_i, f(ab)_{|a|+i}) \\
 &\leq \sum_{i=r+1}^{|a|-r} \delta(a_i, \lambda(a_{i-r}, \dots, a_{i+r})) + r \\
 &\quad + r + \sum_{i=r+1}^{|b|-r} \delta(b_i, \lambda(b_{i-r}, \dots, b_{i+r})) \\
 &\leq \Delta(r|a|_r, f(a)) + 2r + \Delta(r|b|_r, f(b)) .
 \end{aligned}$$

□

En appliquant le lemme précédent plusieurs fois, on peut montrer qu'il est encore vrai avec une concaténation de  $h$  mots :

**Lemme 2.** Soit  $(a_i)_{i \in [1, h]}$  une suite finie de  $h$  mots de taille supérieure à  $2r$ . Soit  $x = a_1 \dots a_h$ , la concaténation des mots de cette suite. Alors, pour tout automate cellulaire  $f$  de rayon  $r$ ,

$$\Delta(r|x|_r, f(x)) \leq \sum_{i=1}^h \Delta(r|a_i|_r, f(a_i)) + 2r(h-1)$$

*Preuve.* Par récurrence sur  $h$ . Pour  $h = 1$ , c'est immédiat.

Supposons que le lemme est vrai pour  $h = n$ , alors, pour  $x = a_1 \dots a_{n+1}$ , si  $x' = a_1 \dots a_n$  (et donc  $x = x'a_{n+1}$ ). On a, d'après le lemme 1 :

$$\Delta(r|x|_r, f(x)) \leq \Delta(r|x'|_r, f(x')) + \Delta(r|a_{n+1}|_r, f(a_{n+1})) + 2r ,$$

et donc, par hypothèse de récurrence,

$$\begin{aligned} \Delta(r|x|_r, f(x)) &\leq \sum_{i=1}^n \Delta(r|a_i|_r, f(a_i)) + 2r(n-1) + \\ &\quad \Delta(r|a_{n+1}|_r, f(a_{n+1})) + 2r \\ &= \sum_{i=1}^{n+1} \Delta(r|a_i|_r, f(a_i)) + 2rn . \end{aligned}$$

□

Nous avons également besoin d'un résultat connu sur les séries :

**Lemme 3 (lemme de Césaro).** *Soit  $(a_n)_{n \in \mathbb{N}}$  et  $(u_n)_{n \in \mathbb{N}}$ , deux séries réelles telles que :*

$$\lim_{n \rightarrow \infty} \frac{u_n}{a_n} = l$$

la série  $a_n$  étant divergente et positive. Alors,

$$\lim_{n \rightarrow \infty} \frac{\sum_{i=0}^n u_i}{\sum_{i=0}^n a_i} = l .$$

A présent, nous allons démontrer divers résultats permettant de décrire les points fixes et périodiques d'un automate cellulaire  $f$  dans la topologie de Besicovitch, c'est-à-dire les points fixes et périodiques de  $\dot{f}$ .

**Proposition 5.** *Dans l'espace de Besicovitch, si un automate cellulaire  $f$  a deux points périodiques distincts de périodes  $p_1$  et  $p_2$ , alors, il a un nombre non dénombrable de points périodiques de période  $\text{ppcm}(p_1, p_2)$ .*

*Preuve.* Soient  $x$  et  $y$  deux points périodiques de  $\dot{f}$ , dont les périodes sont respectivement  $p_1$  et  $p_2$ . Soit  $p = \text{ppcm}(p_1, p_2)$ . Soit  $r$  le rayon de  $f$ . Soient  $x'$  un représentant de la classe  $x$  et  $y'$  un représentant de la classe  $y$ . Comme  $x$  et  $y$  sont deux points distincts de  $\dot{X}$ , on a  $d_{\mathbb{B}}(x', y') = \delta > 0$ . Il existe une suite strictement croissante d'entiers  $(u_n)_{n \in \mathbb{N}}$  telle que pour tout entier  $n > 0$ ,

$$\Delta(x'_{-u_n : u_n}, y'_{-u_n : u_n}) \geq \delta u_n . \quad (1.3)$$

Soit  $\sigma$  une fonction strictement croissante telle que  $u_{\sigma(0)} > 4r$  et  $u_{\sigma(n+1)} > 2u_{\sigma(n)}$ . Soit  $v_n = u_{\sigma(n)}$ . Ainsi, on a :

$$\sum_{i=0}^n v_i \leq 2v_n . \quad (1.4)$$

En effet, par récurrence sur  $n$ , on a  $v_0 \leq 2v_0$ , et si  $2v_{n-1} \geq \sum_{i=0}^{n-1} v_i$  alors  $2v_n \geq v_n + 2v_{n-1} \geq v_n + \sum_{i=0}^{n-1} v_i \geq \sum_{i=0}^n v_i$ .

A présent, nous construisons une injection  $g$  de  $\{0, 1\}^{\mathbb{Z}}$  dans  $X$  (et non  $\dot{X}$ ) telle que toute classe contenant un point de l'ensemble image est un point périodique de  $\dot{f}$ . Soit  $\alpha$  une séquence de  $\{0, 1\}^{\mathbb{N}}$ . Pour tout entier positif  $i$ , soit

$k_i^\alpha = x'_{1:v_i}$  si  $\alpha_i = 0$  et  $y'_{1:v_i}$  si  $\alpha_i = 1$ , et soit  $k_i^{\prime\alpha} = x'_{-v_i:-1}$  si  $\alpha_i = 0$  et  $y'_{-v_i:1}$  si  $\alpha_i = 1$ . On définit  $g$  de la manière suivante :

$$g(\alpha) = \dots k_n^{\prime\alpha} k_{n-1}^{\prime\alpha} \dots k_2^{\prime\alpha} k_1^{\prime\alpha} k_0^{\prime\alpha} 0^{2rp+1} k_0^\alpha k_1^\alpha k_2^\alpha \dots k_{n-1}^\alpha k_n^\alpha \dots$$

Démontrons que la classe de  $g(\alpha)$  est un point périodique de  $f$  de période  $p$ , c'est-à-dire que  $d_B(g(\alpha), f^p(g(\alpha))) = 0$ .

Il faut prouver que :

$$\limsup_{n \rightarrow \infty} \frac{\Delta(g(\alpha)_{-n:n}, f^p(g(\alpha))_{-n:n})}{2n} = 0 .$$

Comme  $x$  et  $y$  sont des points périodiques de périodes respectives  $p_1$  et  $p_2$  qui sont toutes deux des diviseurs de  $p$ , on sait que  $d_B(x', f^p(x')) = d_B(y', f^p(y')) = 0$  et donc que :

$$\lim_{n \rightarrow \infty} \frac{\Delta(x'_{-n:n}, f^p(x')_{-n:n})}{2n} = \lim_{n \rightarrow \infty} \frac{\Delta(y'_{-n:n}, f^p(y')_{-n:n})}{2n} = 0.$$

Par conséquent, on a :

$$\lim_{n \rightarrow \infty} \frac{\Delta(x'_{1:n}, f^p(x')_{1:n})}{n} = \lim_{n \rightarrow \infty} \frac{\Delta(y'_{1:n}, f^p(y')_{1:n})}{n} = 0. \quad (1.5)$$

Fixons un entier  $n > r$  et décomposons  $g(\alpha)_{-n:n}$  en ses facteurs  $k_i^\alpha$  :

$$g(\alpha)_{-n:n} = \overrightarrow{k_h^{\prime\alpha}} k_{h-1}^{\prime\alpha} \dots k_2^{\prime\alpha} k_1^{\prime\alpha} k_0^{\prime\alpha} 0^{2rp+1} k_0^\alpha k_1^\alpha k_2^\alpha \dots k_{h-1}^\alpha \overleftarrow{k_h^\alpha}$$

où  $h$  dépend de  $n$ , et  $\overrightarrow{k_h^{\prime\alpha}}$  [resp.  $\overleftarrow{k_h^\alpha}$ ] est le suffixe [resp. préfixe] de  $k_h^{\prime\alpha}$  [resp.  $k_h^\alpha$ ] situé au début [resp. à la fin] de  $g(\alpha)_{-n:n}$  (Fig. 1.15). En remarquant que  $f^p$

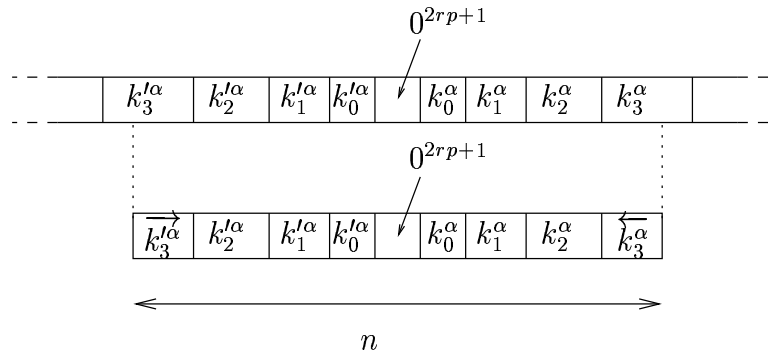


FIG. 1.15 – Illustration de la preuve

est un automate cellulaire de rayon  $rp$ , grâce au lemme 2, on a :

$$\begin{aligned} \Delta(g(\alpha)_{-n:n}, f^p(g(\alpha))_{-n:n}) &\leq \Delta(0, f^p(0^{2rp+1})) \\ &+ \sum_{i=0}^{h-1} \Delta(r_p | k_i^\alpha |_{rp}, f^p(k_i^\alpha)) \\ &+ \sum_{i=0}^{h-1} \Delta(r_p | k_i^{\prime\alpha} |_{rp}, f^p(k_i^{\prime\alpha})) \\ &+ \Delta(r_p | \overrightarrow{k_h^\alpha} |_{rp}, f^p(\overrightarrow{k_h^\alpha})) \\ &+ \Delta(r_p | \overleftarrow{k_h^\alpha} |_{rp}, f^p(\overleftarrow{k_h^\alpha})) + 2rp(2h) . \end{aligned}$$

Par définition de  $k_i^\alpha$ , on sait que  $\overleftarrow{k_h^\alpha}$  est un préfixe de  $x'$  ou de  $y'$ . Par conséquent,

$$\Delta(r_p | \overleftarrow{k_h^\alpha} |_{rp}, f^p(\overleftarrow{k_h^\alpha})) \leq \Delta(r_p | x'_{1:n} |_{rp}, f^p(x'_{1:n})) + \Delta(r_p | y'_{1:n} |_{rp}, f^p(y'_{1:n})) ;$$

de même,

$$\begin{aligned} \Delta(r_p | \overrightarrow{k_h^\alpha} |_{rp}, f^p(\overrightarrow{k_h^\alpha})) &\leq \Delta(r_p | x'_{-n:-1} |_{rp}, f^p(x'_{-n:-1})) \\ &+ \Delta(r_p | y'_{-n:-1} |_{rp}, f^p(y'_{-n:-1})) ; \end{aligned}$$

d'où,

$$\begin{aligned} \Delta(g(\alpha)_{-n:n}, f^p(g(\alpha))_{-n:n}) &\leq 1 + \sum_{i=0}^{h-1} \Delta(r_p | k_i^\alpha |_{rp}, f^p(k_i^\alpha)) \\ &+ \sum_{i=0}^{h-1} \Delta(r_p | k_i^{\prime\alpha} |_{rp}, f^p(k_i^{\prime\alpha})) \\ &+ \Delta(r_p | x'_{-n:n} |_{rp}, f^p(x'_{-n:n})) \\ &+ \Delta(r_p | y'_{-n:n} |_{rp}, f^p(y'_{-n:n})) \\ &+ 4rph \end{aligned} \tag{1.6}$$

On a aussi, d'après l'équation (1.5),

$$\begin{aligned} \lim_{\substack{i \rightarrow \infty \\ \alpha_i = 1}} \frac{\Delta(r_p | k_i^\alpha |_{rp}, f^p(k_i^\alpha))}{v_i} &\leq \lim_{i \rightarrow \infty} \frac{\Delta(r_p | x'_{-v_i:v_i} |_{rp}, f^p(x'_{-v_i:v_i}))}{v_i} \\ &\lim_{n \rightarrow \infty} \frac{\Delta(x'_{-n:n}, f^p(x')_{-n:n})}{n} = 0. \end{aligned}$$

De même :

$$\lim_{\substack{i \rightarrow \infty \\ \alpha_i = 0}} \frac{\Delta(r_p | k_i^\alpha |_{rp}, f^p(k_i^\alpha))}{v_i} = 0.$$

En sommant les deux équations précédentes, on obtient :

$$\lim_{i \rightarrow \infty} \frac{\Delta(r_p | k_i^\alpha |_{rp}, f^p(k_i^\alpha))}{v_i} = 0.$$

Grâce au lemme 3,

$$\lim_{h \rightarrow \infty} \frac{\sum_{i=0}^h \Delta(r_p | k_i^\alpha |_{rp}, f^p(k_i^\alpha))}{\sum_{i=0}^h v_i} = 0,$$



car la série  $(v_i)_{i \in \mathbb{N}}$  est positive et divergente. Le même raisonnement s'applique aux  $k'$  et par conséquent, on obtient

$$\lim_{h \rightarrow \infty} \frac{\sum_{i=0}^h \Delta(r_p | k_i^\alpha |_{r_p}, f^p(k_i^\alpha))}{\sum_{i=0}^h v_i} + \frac{\sum_{i=0}^h \Delta(r_p | k_i'^\alpha |_{r_p}, f^p(k_i'^\alpha))}{\sum_{i=0}^h v_i} = 0 .$$

Or  $\sum_{i=0}^h v_i \leq 2n$ , donc :

$$\lim_{n \rightarrow \infty} \frac{\sum_{i=0}^n \Delta(r_p | k_i^\alpha |_{r_p}, f^p(k_i^\alpha))}{2n} + \frac{\sum_{i=0}^n \Delta(r_p | k_i'^\alpha |_{r_p}, f^p(k_i'^\alpha))}{2n} = 0 \quad (1.7)$$

Toujours grâce à l'équation (1.5), on a :

$$\lim_{n \rightarrow \infty} \frac{\Delta(r_p | x'_{-n:n} |_{r_p}, f^p(x'_{-n:n}))}{2n} = \lim_{n \rightarrow \infty} \frac{\Delta(r_p | y'_{-n:n} |_{r_p}, f^p(y'_{-n:n}))}{2n} = 0 \quad (1.8)$$

Finalement,

$$\lim_{n \rightarrow \infty} \frac{4rph + 1}{2n} = 0 \quad (1.9)$$

puisque  $h \leq \ln_2(n)$ .

En utilisant les équations (1.7), (1.8) et (1.9) dans l'équation (1.6), on obtient :

$$\lim_{n \rightarrow \infty} \frac{\Delta(g(\alpha)_{-n:n} f^p(g(\alpha)_{-n:n})}{2n} = 0$$

et ainsi, que  $g(\alpha)$  est un point périodique de période  $p$  de  $f$ .

Soit  $\sim$ , la relation d'équivalence qui lie deux suites de  $\{0, 1\}^{\mathbb{N}}$  si et seulement si elles ne diffèrent qu'en un nombre fini de points. Soient  $\alpha$  et  $\beta$ , deux suites de  $\{0, 1\}^{\mathbb{N}}$  telles que  $\alpha \approx \beta$ . Soit  $(a_i)_{i \in \mathbb{N}}$  la suite croissante des points où elles diffèrent. Alors,

$$\begin{aligned} d_{\mathbb{B}}(g(\alpha), g(\beta)) &\geq \limsup_{n \in \mathbb{N}} \frac{\Delta(g(\alpha)_{-n:n}, g(\beta)_{-n:n})}{2n} \\ &\geq \limsup_{i \in \mathbb{N}} \frac{\Delta(k_{a_n}^\alpha, k_{a_n}^\beta)}{2 \sum_{i=0}^{a_n} v_i + 2r + 1} \\ &\geq \limsup_{i \in \mathbb{N}} \frac{\Delta(x'_{v_{a_n}}, y'_{v_{a_n}})}{2 \sum_{i=0}^{a_n} v_i + 2r + 1} \end{aligned}$$

En appliquant l'équation (1.3) et (1.4), on obtient :

$$\begin{aligned} d_{\mathbb{B}}(g(\alpha), g(\beta)) &\geq \limsup_{n \in \mathbb{N}} \frac{\delta v_{a_n}}{2v_{a_n} + 1} \\ &\geq \frac{\delta}{2} . \end{aligned}$$

Ainsi,  $g(\alpha)$  et  $g(\beta)$  sont dans des classes différentes.

On vient de prouver que les images par  $g$  sont toutes des points périodiques de période  $p$ , et que :

$$\alpha \approx \beta \implies g(\alpha) \not\approx g(\beta) .$$

Soit  $E$  un ensemble contenant un représentant de chaque classe d'équivalence de  $\sim$ . Comme  $\{0, 1\}^{\mathbb{N}} / \sim$  est indénombrable,  $E$  aussi. D'après l'équation ci-dessus,  $g|_E$  est injective donc  $g(E)$  est un ensemble non dénombrable de points périodiques de période  $p$  de  $f$ .  $\square$

On peut déduire différents corollaires de la proposition précédente :

**Corollaire 1.** *Si un automate cellulaire  $f$  a deux points fixes, alors il admet un nombre non dénombrable de points fixes.*

*Preuve.* Puisqu'un point fixe est un point périodique de période 1, c'est un cas particulier de la proposition 5.  $\square$

**Corollaire 2.** *Si un automate cellulaire a un point périodique de période  $p > 1$ , alors il possède un nombre non dénombrable de points périodiques.*

*Preuve.* Si  $x$  est un point périodique de période  $p > 1$  d'un automate cellulaire  $f$ , il en est de même pour  $f(x)$ , qui est différent de  $x$  puisque  $p > 1$ . Alors, grâce à la proposition 5,  $f$  a un nombre non dénombrable de points périodiques.  $\square$

La proposition suivante est une conséquence immédiate des corollaires 1 et 2.

**Proposition 6.** *Tout automate cellulaire a exactement un point fixe, ou un nombre indénombrable de points périodiques.*

*Preuve.* Soit  $f$  un automate cellulaire. Ou bien  $f$  a exactement un point fixe ; ou bien  $f$  a deux points fixes ou plus, et dans ce cas, il a un nombre non dénombrable de points périodiques (ici, ce sont des points fixes) d'après le corollaire 1 ; ou alors  $f$  n'a pas de points fixes et dans ce cas,  $f(\underline{0}) = \underline{1}$  et  $f(\underline{1}) = \underline{0}$  (par définition d'un automate cellulaire), et, d'après le corollaire 2,  $f$  admet un nombre non dénombrable de points périodiques.  $\square$

**Proposition 7.** *Si un automate cellulaire est surjectif et admet un mot bloquant (ou, de manière équivalente, un point d'équicontinuité pour la topologie de Cantor), alors il a un ensemble dense de points périodiques.*

*Preuve.* Soit  $f$  un automate cellulaire et  $w$  un mot bloquant de  $f$ .

Il faut prouver que pour toute configuration  $x$ , et pour tout  $\varepsilon > 0$ , il existe une configuration périodique à distance moindre que  $\varepsilon$  de  $x$ .

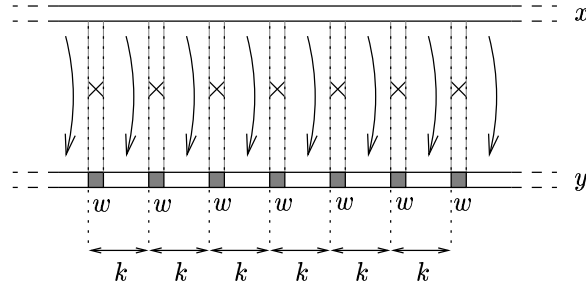
Soit  $y$ , la configuration suivante :

$$\forall n, l < k \in \mathbb{N}, y_{nk+l} = \begin{cases} w_l & \text{if } l < |w| \\ x_{nk+l} & \text{sinon} \end{cases}$$

où  $k = \lceil 2\varepsilon|w| \rceil$ . La configuration  $y$  est partout égale à la configuration  $x$  sauf à des endroits réguliers où l'on met  $w$  à la place (Fig. 1.16).

Le nombre de différences entre  $x_{-n:n}$  et  $y_{-n:n}$  est donc borné par le nombre de fois où  $w$  est écrit multiplié par  $|w|$ . On a donc :

$$\begin{aligned} d_{\mathcal{B}}(x, y) &\leq \lim_{n \rightarrow \infty} \frac{\Delta(x_{-n:n}, y_{-n:n})}{2n+1} \\ &\leq \lim_{n \rightarrow \infty} \frac{\lceil \frac{2n}{k} \rceil |w|}{2n} \\ &\leq \lim_{n \rightarrow \infty} \frac{\frac{n}{\varepsilon} + |w|}{2n} \\ &< \varepsilon . \end{aligned}$$

FIG. 1.16 – Construction de la figure  $y$  en fonction de  $k = 2\varepsilon|w|$  et  $x$ 

Montrons que  $y$  est un point périodique pour  $f$ . Comme  $w$  est un mot bloquant, pour tous entiers  $i$  et  $n$ , le motif  $f^{i+1}(y)_{nk+\lceil |w|/2 \rceil : n(k+1)+\lceil |w|/2 \rceil}$  ne dépend que du motif de même taille correspondant dans l'antécédent, à savoir  $f^i(y)_{nk+\lceil |w|/2 \rceil : n(k+1)+\lceil |w|/2 \rceil}$ . Autrement dit les zones grisées de la figure 1.17 sont indépendantes les unes des autres. Pour tous entiers  $i$  et  $n$ , posons

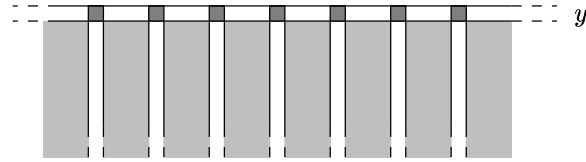


FIG. 1.17 – Les zones grisées sont indépendantes les unes des autres

$u_n^{(i)} = f^i(y)_{nk+\lceil |w|/2 \rceil : n(k+1)+\lceil |w|/2 \rceil}$ . Pour tout  $n$  fixé, la suite  $(u_n^{(i)})_{i \in \mathbb{N}}$  est une suite à valeur bornée, dont le  $i^{\text{ème}}$  terme ne dépend que du précédent. Par conséquent, elle est ultimement périodique. Cependant, comme l'automate cellulaire de départ est surjectif, elle est en fait périodique (sinon, si la suite n'est périodique de période  $p$  qu'au rang  $k$ , alors  $u_n^{(k)}$  a deux antécédents :  $u_n^{(k-1)}$  et  $u_n^{(k+p-1)}$ ). Comme le nombre de mots de longueur inférieure à  $k$  est  $2^k$ , la période de chaque colonne est donc au plus  $2^p$ . La configuration  $y$  est donc périodique, de période inférieure ou égale à  $\text{ppcm}\{2^i, 1 \leq i \leq k\}$ .  $\square$

### 1.2.2.2 Séquences de Toeplitz

Dans la topologie de Cantor, les configurations spatialement périodiques jouent un rôle particulier. En effet, l'ensemble des configurations spatialement périodiques est dense. Ainsi, il est souvent équivalent de montrer une propriété pour un automate cellulaire et pour cet automate, restreint à ces configurations ([Knu94]).

Dans la topologie de Besicovitch, l'ensemble des configurations spatialement périodiques n'est pas dense. En effet, si on considère la configuration  $c$  suivante, en prenant deux états 0 et 1 :

$$c_i = \begin{cases} 0 & \text{si } i < 0 \\ 1 & \text{sinon} \end{cases}$$

alors toute configuration spatialement périodique est à distance supérieure à  $\frac{1}{2}$  de  $c$ . En effet, soit  $x$ , une configuration périodique de période  $p$ . Supposons qu'un motif périodique de taille  $p$  contienne  $q$  cellules différentes de 0 et donc  $p - q$  cellules dont l'état est 0 (cela ne dépend pas du motif périodique choisi). On a donc :

$$\Delta(x_{-n:n}, c_{-n:n}) = \#_{\neq 0}(x_{-n:n}) + \#_{\neq 1}(x_{0:n}) \geq q \left\lfloor \frac{n}{p} \right\rfloor + (p - q) \left\lfloor \frac{n}{p} \right\rfloor = p \left\lfloor \frac{n}{p} \right\rfloor$$

où  $\#_{\neq i}(m)$  est le nombre de cellules du motif  $m$  dont l'état n'est pas  $i$ . Par conséquent :

$$d_B(x, c) = \limsup_{n \rightarrow \infty} \frac{\Delta(x_{-n:n}, c_{-n:n})}{2n + 1} \geq \limsup_{n \rightarrow \infty} \frac{p \left\lfloor \frac{n}{p} \right\rfloor}{2n + 1} = \frac{1}{2}$$

Nous allons donc étudier le comportement des automates cellulaires sur l'ensemble de configurations de Toeplitz, dans le cadre de la topologie de Besicovitch, et ainsi montrer le parallèle entre ces dernières et les configurations périodiques dans la topologie de Cantor.

**Définition 17 (configurations de Toeplitz).**

Une configuration  $x$  est une séquence de Toeplitz ou une configuration de Toeplitz si pour tout entier relatif  $i$ , il existe une période  $p_i$  telle que, pour tout  $k \in \mathbb{Z}$ ,  $x_{p_i k + i} = x_i$ . L'état de toute cellule est répété périodiquement, la période dépendant de la cellule. On notera  $\mathcal{T}$  l'ensemble des configurations de Toeplitz.

Remarquons que tout motif dans une séquence de Toeplitz est périodique, la période est le ppcm des périodes des caractères qui le composent.

**Lemme 4.** La restriction de la surjection canonique  $\Pi$  de  $X$  vers  $\hat{X}$  aux configurations de Toeplitz est injective.

*Preuve.* Soient  $x$  et  $y$  deux configurations de Toeplitz distinctes de  $X$ , et qui diffèrent, donc, au moins à une position  $\alpha$ . Par définition, il existe un entier  $u$  tel que pour tout  $k \in \mathbb{Z}$ , ces configurations diffèrent à la position  $\alpha + uk$ . Par conséquent la distance entre ces configurations est supérieure à  $u^{-1}$ . Elles ne sont donc pas dans la même classe, ce qui signifie que leurs images par  $\Pi$  sont distinctes.  $\square$

On vient de prouver que dans une classe d'équivalence, il y a au plus une configuration de Toeplitz. Cependant, certaines classes n'en contiennent pas. Si l'on reprend la configuration  $c$  :

$$c_i = \begin{cases} 0 & \text{si } i < 0 \\ 1 & \text{sinon} \end{cases}$$

alors pour toute configuration de Toeplitz  $x$ , si la période de  $x_0$  est  $p$ ,  $d(x, c) \geq \frac{1}{2p}$ , car les répétitions périodiques de  $x_0$  vers la gauche si  $x_0 \neq 0$  ou vers la droite si  $x_0 = 0$  induisent autant d'erreurs.

**Lemme 5.** Pour tout automate cellulaire  $f$ , l'ensemble des configurations de Toeplitz est invariant; autrement dit, l'image d'une configuration de Toeplitz par un automate cellulaire est une configuration de Toeplitz.

*Preuve.* Soit  $x \in \mathcal{T}$ , et  $f$  un automate cellulaire de rayon  $r$  et de règle locale  $\lambda$ . Pour tous  $i$  et  $k$  de  $\mathbb{Z}$ , il existe  $p_i$  tel que  $x_{p_i k+i} = x_i$ . Soit  $p'_i = \text{ppcm}_{-r \leq l \leq r} \{p_{i+l}\}$ . Alors on a :

$$f(x)_i = \lambda(x_{i-r}, \dots, x_{i+r}) = \lambda(x_{p'_i k+i-r}, \dots, x_{p'_i k+i+r}) = f(x)_{p'_i k+i-r} .$$

La configuration  $f(x)$  est une configuration de Toeplitz.  $\square$

**Proposition 8.** *L'ensemble des configuration de Toeplitz est dense pour la topologie de Besicovitch.*

*Preuve.* Soient  $x$  une configuration de Toeplitz et  $\varepsilon > 0$ . Construisons une configuration de Toeplitz à distance moindre que  $\varepsilon$  de  $x$ .

Soit  $(n_k)_{k \in \mathbb{N}}$  une suite d'entiers satisfaisant les conditions suivantes :

- i) pour tous entiers  $p$  et  $q$  tels que  $p < q$  alors  $n_p$  divise  $n_q$  ;
- ii)  $\sum_{i=0}^{\infty} n_i^{-1}$  converge vers une limite inférieure à  $\varepsilon$ .

Le processus de construction de la configuration  $y$  est récursif. L'étape d'initialisation est la suivante : pour tout  $i \in \mathbb{Z}$  affecter  $y_{i n_0} = x_0$ . Ensuite, pour  $k$ , entier croissant de 1 à l'infini, on effectue les deux étapes suivantes :

Étape  $k - \frac{1}{2}$  : si  $y_k$  n'a pas déjà été affecté, assigner, pour tout  $i \in \mathbb{Z}$ ,

$$y_k = y_{k+i n_{2k-1}} = x_k ;$$

Étape  $k$  : si  $y_{-k}$  n'a pas déjà été affecté, assigner, pour tout  $i \in \mathbb{Z}$ ,  $y_{-k} =$

$$y_{-k+i n_{2k}} = x_k .$$

Posons  $u_k = -k$  si  $k$  est entier, et  $u_k = k + \frac{1}{2}$  sinon. La position  $u_k$  est la position concernée par le test de l'étape  $k$ .

Tout d'abord, il faut prouver que la construction est correcte, et qu'aucune des cellules de  $y$  n'est affectée deux fois. Supposons par l'absurde que cette éventualité se produise et que la cellule  $m$  de  $y$  soit affectée deux fois, d'abord à l'étape  $a$  puis à l'étape  $b$  ( $a < b$ ). Dans ce cas, il existe deux entiers  $k_1$  et  $k_2$  tels que  $m = u_a + k_1 n_{2a} = u_b + k_2 n_{2b}$ . La position  $u_b$  n'était pas assignée à l'étape  $b$ , sinon aucun assignement n'aurait été effectué pendant cette étape. D'après la condition i) sur la suite  $n$ , il existe un entier  $p$  tel que  $n_{2b} = p n_{2a}$  d'où  $u_b = u_a + (k_1 - p k_2) n_{2a}$ . Par conséquent la position  $u_b$  a été affectée à l'étape  $a$  et l'était donc déjà à l'étape  $b$ , ce qui est absurde. De plus, par construction, toutes les cellules sont bien affectées. La configuration  $y$  est donc bien une séquence de Toeplitz, puisque toute cellule a pour période  $n_{2a}$  si  $a$  est l'étape à laquelle elle a été assignée.

Montrons que  $d(x, y) < \varepsilon$ . Majorons le nombre  $N(n)$  de positions  $i$  entre  $-n$  et  $n$  où  $x_i$  et  $y_i$  diffèrent. L'entier  $N(n)$  est inférieur au nombre de fois où la position  $u_b$  entre  $-n$  et  $n$  était déjà assignée à l'étape  $b$ . En effet, dans le cas contraire, on affecte  $x_{u_b} = y_{u_b}$ . Entre  $-n$  et  $n$ , il y a au plus  $\left\lfloor \frac{2n+1}{n_{2b}} \right\rfloor + 1$  positions affectées à l'étape  $b$ ; comme, parmi ces positions,  $u_b$  est affectée telle que  $x_{u_b} = y_{u_b}$ , le nombre d'erreurs commises à l'étape  $b$  est au plus  $\left\lfloor \frac{2n+1}{n_{2b}} \right\rfloor$ . On en déduit que :

$$N(n) \leq \sum_{b \in \frac{1}{2}\mathbb{N}} \left\lfloor \frac{2n+1}{n_{2b}} \right\rfloor \leq \sum_{i \in \mathbb{N}} \frac{2n+1}{n_i} \leq (2n+1) \sum_{i \in \mathbb{N}} n_i^{-1} \leq (2n+1)\varepsilon$$

grâce à la condition ii) sur la suite  $n$ . En divisant cette équation par  $2n+1$  et faisant tendre  $n$  vers l'infini, on déduit que  $d(x, y) \leq \varepsilon$ .  $\square$

**Proposition 9** ([BFK99]). *Soit  $f$  un automate cellulaire. La fonction  $f$  est surjective si et seulement si la fonction  $\hat{f}$  est surjective.*

Grâce cette proposition, on peut démontrer :

**Proposition 10.** *Soit un automate cellulaire  $f$ . La fonction  $\hat{f}$  est surjective si sa restriction aux configurations de Toeplitz est surjective.*

*Preuve.* Si la restriction de  $\hat{f}$  aux configurations de Toeplitz est surjective, alors, en particulier, pour toute configuration spatialement périodique  $x$ , la classe  $\hat{x}$  admet un antécédent  $\hat{y}$  qui contient une configuration de Toeplitz  $y$ . D'après le lemme 5,  $y$  a pour image une configuration de Toeplitz qui se trouve dans  $\hat{x}$ . D'après le lemme 4, il n'y a qu'une seule configuration de Toeplitz par classe. On a donc  $f(y) = x$ . La configuration  $y$  admettant un antécédent, elle admet un antécédent spatialement périodique. Cela implique que la restriction de  $f$  aux configurations spatialement périodiques est surjective. Il est bien connu que cela implique que  $f$  est surjective. Grâce à la proposition 9, on en déduit que  $\hat{f}$  est surjective. □

Cependant, la réciproque de ce théorème reste ouverte :

**Problème Ouvert 1.** *Quel que soit l'automate cellulaire surjectif  $f$ , toute configuration de Toeplitz admettant un antécédent par  $f$  admet-elle un antécédent de Toeplitz ? Est-ce vrai si  $f$  n'est pas supposé surjectif ?*

**Proposition 11.** *Soit  $f$  un automate cellulaire. Si la fonction  $\hat{f}$  est injective alors elle est surjective.*

*Preuve.* Si  $\hat{f}$  est injective, alors, la restriction de  $f$  aux configurations spatialement périodiques est aussi injective. En effet, si on a deux configurations spatialement périodiques  $x$  et  $y$ , les classes  $\hat{x}$  et  $\hat{y}$  sont distinctes (lemme 4) et ont pour images des classes distinctes ; donc les images de  $x$  et  $y$  sont distinctes et spatialement périodiques. Ceci implique que  $f$  est surjective (voir par exemple [Dur98]) et ainsi que  $\hat{f}$  est surjective, grâce à la proposition 9. □

Cet ensemble de résultats prouve que les configurations de Toeplitz dans la topologie de Besicovitch jouent un rôle parallèle aux configurations spatialement périodiques dans la topologie de Cantor.

### 1.2.2.3 Habitation des classes

La question que l'on se pose maintenant est de savoir si les classes sont habitées. En effet, si le changement de topologie résout les paradoxes constatés concernant la chaotité d'automates intuitivement simples, on se demande quelles classes subsistent.

**Exemple 4 (automate cellulaire sensible).** *L'automate cellulaire déjà donné comme exemple 3 est sensible aux conditions initiales ([CFMM97]). Cet automate a pour rayon 2 et pour règle locale  $\lambda(x_{-1}, x_0, x_1) = x_{-1} \oplus x_1$ , ou  $\oplus$  est l'addition modulo 2.*

**Théorème 3.** *Dans l'espace topologique de Besicovitch, il n'y a pas d'automate cellulaire transitif.*

Ce théorème ferme le problème évoqué à plusieurs reprises dans de nombreux articles : [BFK99, CFMM97, Man98, DFM01].

La preuve de ce théorème nécessite la notion de complexité algorithmique (ou complexité de Kolmogorov) qui est définie dans le chapitre suivant, section 2.1.

*Preuve.* Par l'absurde, supposons qu'il existe un automate cellulaire transitif  $f$ , de rayon  $r$  avec  $C$  états. Soient  $x$  et  $y$  deux configurations telles que :

$$\forall n, K(x_{-n:n}|y_{-n:n}) \geq \frac{n}{2} .$$

Elles existent par un simple argument de comptage. Comme  $f$  est transitif, il existe deux configurations  $x'$  et  $y'$  telles que :

$$\begin{aligned} \Delta(x_{-n:n}, x'_{-n:n}) &\leq 4\varepsilon n \\ \Delta(y_{-n:n}, y'_{-n:n}) &\leq 4\delta n \end{aligned} \quad (1.10)$$

et un entier  $u$  (qui ne dépend que de  $\varepsilon$  et  $\delta$ ) tel que :

$$f^u(y') = x', \quad (1.11)$$

où  $\varepsilon = \delta = (4e^{10C})^{-1}$ .

Dans la suite de la preuve, seul  $n$  varie, et  $C$ ,  $u$ ,  $x$ ,  $y$ ,  $x'$ ,  $y'$ ,  $\delta$  et  $\varepsilon$  sont fixés et indépendants de  $n$ . Grâce à l'équation (1.11), on peut calculer le motif  $x'_n$  en connaissant les éléments suivants :

- $y'_{-n:n}$ ,  $f$ ,  $u$  et  $n$  ;
- les deux fois  $ur$  bits de  $y'$ , qui entourent  $y'_{-n:n}$  et qui manquent pour calculer  $x'_{-n:n}$  à l'aide de (1.11).

On en déduit donc que :

$$K(x'_{-n:n}|y'_{-n:n}) \leq 2ur + K(u) + K(n) + K(f) + O(1) \leq o(n) \quad (1.12)$$

(les notations  $O$  et  $o$  sont à considérer ayant comme variable  $n$ ).

Essayons ensuite d'évaluer  $K(y'_{-n:n}|y_{-n:n})$ . Soient  $a_1, a_2, a_3, \dots, a_k$  les positions positives auxquelles  $y_{-n:n}$  et  $y'_{-n:n}$  diffèrent, classées par ordre croissant. Soient  $b_1 = a_1$  et  $b_i = a_i - a_{i-1}$  pour  $2 \leq i \leq k$ . Grâce à l'équation (1.10), on sait que  $k \leq 4\delta n$ . On remarque que  $\sum_{i=1}^k b_i = a_k \leq n$ .

De même, soient  $a'_1, a'_2, a'_3, \dots, a'_{k'}$  les valeurs absolues des positions strictement négatives auxquelles  $y_{-n:n}$  et  $y'_{-n:n}$  diffèrent, classées par ordre croissant. Soient  $b'_1 = a'_1$  et  $b'_i = a'_i - a'_{i-1}$  pour  $2 \leq i \leq k'$ . Toujours à l'aide de l'équation (1.10), on a  $k' \leq 4\delta n$ .

Comme la fonction logarithme est concave, on a :

$$\sum_{i=1}^k \frac{\ln b_i}{k} \leq \ln \frac{\sum_{i=1}^k b_i}{k} \leq \ln \frac{n}{k}$$

et donc :

$$\sum_{i=1}^k \ln b_i \leq k \ln \frac{n}{k} \quad (1.13)$$

qui est aussi vraie avec  $b'_i$  et  $k'$ .

La connaissance des  $b_i$ , des  $b'_i$ , et des  $C(k+k')$  bits de  $y'_{-n:n}$  aux positions où  $y_{-n:n}$  diffère de  $y'_{-n:n}$  est suffisante pour calculer  $y'_{-n:n}$  à partir  $y_{-n:n}$ . Ainsi :

$$K(y'_{-n:n}|y_{-n:n}) \leq \sum \ln(b_i) + \sum \ln(b'_i) + C(k+k') + O(1) .$$

Grâce à l'équation (1.13) on a :

$$K(y'_{-n:n}|y_{-n:n}) \leq k \ln \frac{n}{k} + k' \ln \frac{n}{k'} + C(k+k')O(1) .$$

La fonction  $k \mapsto k \ln \frac{n}{k}$  est croissante sur  $[0, \frac{n}{e}]$ . Comme  $k \leq 4\delta n \leq \frac{n}{e^{10C}}$ , on obtient que :

$$k \ln \frac{n}{k} \leq 4\delta n \ln \frac{n}{4\delta n} \leq \frac{4n}{e^{10C}} \ln e^{10C} \leq \frac{40}{e^{10C}} n$$

et que :

$$C(k+k') \leq \frac{2Cn}{e^{10C}} .$$

Grâce à la même suite d'inégalités, mais en substituant  $a, b$  et  $k$  par  $a', b'$  et  $k'$  on aboutit au même résultat. D'où,

$$K(y'_{-n:n}|y_{-n:n}) \leq \frac{(2C+80)n}{e^{10C}} + O(1) . \quad (1.14)$$

De façon similaire, l'équation (1.14) est aussi vérifiée par  $K(x_{-n:n}|x'_{-n:n})$ .

L'inégalité triangulaire (lemme 6) de la complexité de Kolmogorov ([LV97]) appliquée ici donne :

$$\begin{aligned} K(x_{-n:n}|y_{-n:n}) &\leq K(x_{-n:n}|x'_{-n:n}) + K(x'_{-n:n}|y'_{-n:n}) \\ &\quad + K(y'_{-n:n}|y_{-n:n}) + O(1) . \end{aligned}$$

Grâce à (1.14) et (1.12) on en déduit :

$$K(x_{-n:n}|y_{-n:n}) \leq \frac{(2C+80)n}{e^{10C}} + o(n) .$$

L'hypothèse sur  $y$  et  $x$  était que  $K(x_{-n:n}|y_{-n:n}) \geq \frac{n}{2}$ , et implique

$$\frac{n}{2} \leq \frac{(2C+80)n}{e^{10C}} + o(n)$$

ce qui est absurde pour  $n$  assez grand. Donc notre automate initial n'était pas transitif.  $\square$





## Chapitre 2

# Approche algorithmique à l'étude des Automates Cellulaires et du Chaos

Ce chapitre<sup>1</sup> propose une autre approche pour résoudre les problèmes des définitions de chaotité des automates cellulaires. Nous allons utiliser pleinement la notion de complexité algorithmique pour obtenir une topologie qui soit le plus liée possible à la notion de calculabilité. En effet, mesurer la complexité algorithmique des objets mathématiques n'est certes pas simple, et délicat à manipuler, mais peut être très fructueux pour donner des idées nouvelles ou conduire à de nouvelles applications.

Ici, nous proposons une nouvelle manière d'étudier la complexité des automates cellulaires, en utilisant cette notion pour comparer deux configurations d'un point de vue algorithmique, et obtenir ainsi un *distance algorithmique* sur l'espace des configurations. Nos espoirs sont de trouver une définition robuste, qui soit propre à l'étude des automates cellulaires en tant que systèmes dynamiques. On pourra ainsi avoir une vision plus algorithmique de ce que représente le chaos pour les automates cellulaires.

Une approche dans le même esprit est proposée par Calude *et al* dans [CHJW00]. Des résultats similaires aux nôtres, mais plus faibles, sont donnés : la préservation de l'incompressibilité par les automates cellulaires surjectifs, sa destruction par les automates non surjectifs, et la préservation de la non-compressibilité par tout automate. Cependant, dans cet article, le dernier point est ouvert pour les automates cellulaires de dimension supérieure à 2, alors que nos preuves s'étendent à toute dimension supérieure; et leur preuves ne fonctionnent que dans un espace séparable. Néanmoins, bien des espaces topologiques intéressants, dans lesquels on étudie les systèmes dynamiques, ne sont pas séparables, comme les topologie de Besicovitch ou de Weyl.

---

<sup>1</sup>Ce chapitre a été l'objet de la publication [CDF01]

## 2.1 Complexité Algorithmique

La complexité algorithmique a été découverte indépendamment, de façon informelle par Solomonoff dans [Sol64], de manière plus formelle par Kolmogorov dans [Kol65], puis par Chaitin dans [Cha66].

Dans ce chapitre, on s'intéresse à la complexité des configurations vues précédemment, c'est-à-dire des séquences bi-infinies de symboles choisis dans un alphabet fini (généralement  $\{0, 1\}$ ). Dans ce contexte, la complexité est mesurée en utilisant la définition introduite par Kolmogorov [Kol65] puis par Chaitin [Cha66]. Une présentation moderne du sujet est donnée dans [LV97, Cal94].

On ne considère que les configurations sur l'alphabet  $\{0, 1\}$ . On se fixe une bijection récursive  $x, y \mapsto \langle x, y \rangle$  entre  $(\{0, 1\}^*)^2$  et  $\{0, 1\}^*$ .

**Notation.** Comme dans le chapitre précédent, pour toute configuration  $x$ , on notera  $x_i = x(i)$  la  $i^{\text{ème}}$  coordonnée de  $x$ . On notera  $x_{a:b}$  le mot  $x_a x_{a+1} \dots x_b$  si  $b \geq a$ . Dans le cas d'un mot  $u$ , on notera sa taille  $|u|$ ,  $u_1$  sa première lettre,  $u_i$  sa  $i^{\text{ème}}$  lettre. Enfin, si  $\{0, 1\} \subset X$ , on note  $\underline{0}$  la configuration qui ne contient que des 0 et  $\underline{1}$  celle qui ne contient que des 1.

Soit  $w$ , un mot (de longueur finie) sur l'alphabet  $\{0, 1\}$ . La complexité de  $w$  est la taille d'un plus petit programme qui calcule  $w$ . On dit qu'un programme  $p$  calcule un mot  $w$  sur l'entrée  $z$ , si  $\varphi(\langle p, z \rangle) = w$ , où  $\varphi$  est une fonction calculable (partielle ou totale). On appelle une telle fonction  $\varphi$  un *mode de description*. Ainsi,  $p$  est la plus petite description pour  $w$ , quand  $z$  est donné (voir [US96]).

**Définition 18.** Soient  $x, y$  et  $p$  des mots sur  $\{0, 1\}$ , et  $\varphi$  une fonction calculable de  $\{0, 1\}^*$  dans  $\{0, 1\}^*$ . La complexité de Kolmogorov  $K_\varphi$  de  $x$  sachant  $y$ , relativement au mode de description  $\varphi$  est :

$$K_\varphi(x|y) = \min\{|p|, \varphi(\langle y, p \rangle) = x\},$$

et  $K_\varphi(x|y) = \infty$  si  $p$  n'existe pas, la fonction  $x, y \mapsto \langle x, y \rangle$  étant une bijection calculable donnée entre  $(\{0, 1\}^*)^2$  et  $\{0, 1\}^*$ .

Il peut sembler que la définition précédente manque de robustesse puisqu'elle dépend du mode de description de référence. Le résultat suivant montre comment supprimer cet inconvénient, puisqu'il prouve l'existence d'un mode de description *optimal*, c'est-à-dire qui décrit chaque mot d'une manière plus courte que tout autre (à une constante additive près). Formalisons tout d'abord la notion d'optimalité.

**Définition 19 (Mode de description additivement optimal).**

Un mode de description  $\varphi$  est additivement optimal, si pour tout mode de description  $\psi$  il existe une constante  $c$ , ne dépendant que de  $\psi$  telle que pour tous mots  $x$  et  $y$ ,

$$K_\varphi(x|y) \leq K_\psi(x|y) + c.$$

**Théorème 4 (théorème d'optimalité (Kolmogorov-Solomonoff)).**

Il existe un mode de description additivement optimal.

*Preuve.* Pour tous mots  $x = x_1 \dots x_n$  et  $y$ , notons :

$$\langle x, y \rangle = x_1 x_1 x_2 x_2 \dots x_n x_n 01y \ .$$

Ce codage de deux mots a deux propriétés. Tout d'abord, il permet, à partir de  $\langle x, y \rangle$ , de calculer  $x$  et  $y$ . Ensuite,  $|\langle x, y \rangle| = 2|x| + 2 + |y|$ .

Soit  $\varphi$  le mode de description suivant : si  $\varphi_u$  est la fonction calculée par une machine de Turing universelle,

$$\varphi(\langle \langle x, y \rangle, z \rangle) = \varphi_u(\langle x, \langle y, z \rangle \rangle)$$

Le mode de description  $\varphi$  est additivement optimal. En effet, pour tout mode de description  $\psi$ , il existe un entier  $a$  tel que pour tout  $x$ ,  $\psi(x) = \varphi_u(\langle a, x \rangle)$ . Posons  $c = 2|a| + 2$ . Soient  $x$  et  $y$  deux mots. Si  $p$  est un programme tel que  $K_\psi(x|y) = |p|$  et  $\psi(\langle p, y \rangle) = x$ , alors  $p' = \langle a, p \rangle$  vérifie :

$$\varphi(\langle p', y \rangle) = \varphi_u(a, \langle p, y \rangle) = \psi(\langle p, y \rangle) = x$$

et donc :

$$K_\varphi(x|y) \leq |p'| = 2|a| + 2 + |p| = K_\psi(x|y) + c .$$

□

Quoique les notions de mode de description additivement optimal et de machine universelle soient liées, elles diffèrent cependant, car la première impose en plus que le codage de l'entrée soit efficace.

On peut donc définir, asymptotiquement, de manière robuste la notion de *complexité algorithmique*.

**Définition 20 (complexité algorithmique).** Soit  $\varphi_0$  un mode de description additivement optimal, fixé comme référence. On définit la complexité algorithmique conditionnelle  $K$  de  $x$  sachant  $y$  par  $K(x|y) = K_{\varphi_0}(x|y)$ .

La complexité algorithmique  $K$  d'un mot  $x$  est  $K(x) = K(x|\varepsilon)$ , où  $\varepsilon$  est le mot vide.

La description d'un mot peut être interprétée comme la version compressée de ce mot, dont le mode de description est le programme décompressant. C'est pour cela que les notions de compression et de complexité algorithmique sont intimement liées. Cependant, même si le mode de description optimal, et donc la décompression, sont calculables, ce n'est, en revanche, pas le cas de la compression. On peut juste l'approximer par valeurs supérieures, tout simplement en essayant tous les programmes possibles. La complexité algorithmique n'est donc pas un moyen pratique de compresser des données, mais un outil formel. Un exemple de son utilisation est donné dans la preuve du théorème 3. Un panorama d'autres applications peut être trouvé dans le chapitre 6 de [LV97] montrant ainsi la puissance de cette méthode.

Ces exemples utilisent une notion de mots incompressibles. Kolmogorov a proposé de définir un mot *c-incompressible* si  $|w| - K(w)$  est inférieure à  $c$ . Martin-Löf a défini dans [ML66] qu'un mot est *c'-incompressible* si et seulement s'il ne peut passer aucun test calculable avec un niveau de précision  $c'$  (voir, par exemple, [LV97] page 129). Par la suite, il prouve l'équivalence entre sa définition et celle de Kolmogorov. Les résultats issus de cette théorie sont utilisés dans la plupart des preuves.

Dans certaines preuves, on montre qu'une propriété est absurde avec un mot ou une configuration incompressible. Dans d'autres, on utilise le fait que presque toutes les configurations sont incompressibles. Ainsi, quand on arrive à

démontrer qu'une propriété constructive est vraie pour presque toutes les configurations, et, si le fait de vérifier la propriété est une caractérisation suffisante pour comprimer une configuration, il s'ensuit que presque toutes les configurations sont incompressibles, ce qui est absurde.

On utilise deux définitions d'incompressibilité dans le cadre des configurations. Elles sont moins restrictives que les définitions usuelles, mais ont l'avantage d'imposer une condition à toute taille du sous-motif considéré.

**Définition 21 (configuration log-incompressible).** *Une configuration  $x$  est dite log-incompressible s'il existe une constante  $c$  telle que  $\forall n, K(x_{-n:n}) \geq 2n - c \log n$ .*

Une définition dans laquelle on aurait remplacé  $K(x_{-n:n}) \geq 2n - c \log n$  par  $K(x_{-n:n}) \geq 2n - c$  n'est pas admissible car dans ce cas, il n'existe pas de configuration incompressible (voir [LV97], section 2.5.1, et en particulier le corollaire 2.5.1).

**Définition 22 (configuration faiblement incompressible).**

*Une configuration  $x$  est dite faiblement incompressible si*

$$\limsup_{n \rightarrow \infty} \frac{K(u_{-n:n})}{2n + 1} = 1.$$

Remarquons que l'ensemble des configurations faiblement incompressibles contient strictement l'ensemble des séquences aléatoires ([LV97] page 144) relativement à la mesure de Bernoulli  $\mu$ . Ainsi,  $\mu$ -presque toutes les configurations sont faiblement incompressibles.

### 2.1.1 Lemmes techniques sur la complexité algorithmique

Lors des preuves utilisant la complexité algorithmique, nous avons besoin de lemmes techniques classiques.

**Lemme 6 (Inégalité triangulaire).** *Il existe une constante  $c$  telle que, pour tous mots  $x, y$  et  $z$  :*

$$K(x|z) \leq K(x|y) + K(y|z) + 2 \min(K(K(x|y)), K(K(y|z))) + c$$

et donc :

$$K(x|z) \leq K(x|y) + K(y|z) + 2 \log \min(|x|, |y|) + c$$

**Notation.** *Si  $n$  est un entier, on note  $\underline{n}$  le mot égal à la représentation binaire de  $n$ . Pour tout mot  $a$  de longueur  $\ell$ , on note  $\bar{a}$  le codage autodélimité suivant :*

$$\bar{a} = 1^{\ell} 0 \underline{\ell} a .$$

*Ce codage est dit autodélimité car, quels que soient les mots  $a$  et  $b$ , à partir du mot  $\bar{a}\bar{b}$ , on peut calculer  $a$  et  $b$ . On a  $|\bar{a}| = |a| + \lceil \log |a| \rceil + 1$ .*

*Preuve.* Soit  $\lambda a, b \}$  le codage de deux entiers suivant :

$$\lambda a, b \} = \begin{cases} 1\bar{a}\bar{b} & \text{si } |a| \leq |b| \\ 0\bar{b}\bar{a} & \text{si } |b| < |a|. \end{cases}$$

Comme pour la preuve de l'existence d'un mode additivement optimal, ce codage a deux propriétés. Tout d'abord on peut calculer  $a$  et  $b$  et fonction  $\langle a, b \rangle$ , et ensuite :

$$|\langle a, b \rangle| \leq |a| + |b| + \lceil \min(\log |a|, \log |b|) \rceil + 2$$

Soit  $\varphi$  le mode de description qui à  $\langle \langle a, b \rangle, c \rangle$  associe :

$$\varphi_u(\langle a, \varphi_u(\langle b, c \rangle) \rangle)$$

où  $\varphi_u$  est la fonction calculée par une machine Turing universelle. Si  $x$ ,  $y$  et  $z$  sont trois mots, et  $p$  est un programme de taille  $K(x|y)$  qui calcule  $x$  en fonction de  $y$ , et  $q$  un programme de taille  $K(y|z)$  qui calcule  $y$  en fonction de  $z$ , alors, en posant  $l = \langle p, q \rangle$ , on a :

$$\varphi(\langle l, z \rangle) = \varphi_u(\langle p, \varphi_u(\langle q, z \rangle) \rangle) = \varphi_u(\langle p, y \rangle) = x .$$

Grâce au théorème d'optimalité, il existe une constante  $c$  telle que

$$K(x|z) \leq |l| + c \leq K(x|y) + K(y|z) + 2 \min(K(K(x|y)), K(K(y|z))) + c$$

□

**Notation.** Si  $A$  est un ensemble fini, on notera  $x/A$  le mot formé des bits de  $x$  dont les positions sont dans  $A$ . Ainsi, si  $A = \{a_1, \dots, a_n\}$ , avec  $a_1 < \dots < a_n$ ,  $x/A = x_{a_1} \dots x_{a_n}$ . On notera  $x \parallel_A^n$  le mot  $w$  de taille  $2n + 1$  avec  $w_i = x_{i-n}$  si  $i \in A$  et  $w_i = 0$  sinon.

Dans les lemmes suivants, on est amené à considérer la complexité algorithmique d'un ensemble. Si  $A$  est un ensemble fini, on notera  $K(A)$  la complexité d'un mot représentant l'énumération des éléments de  $A$ . La représentation de l'ensemble doit être fixé à l'avance et une fois pour toute. Une représentation doit être une injection récursive entre l'ensemble des parties de  $\mathbb{N}$  et l'ensemble des mots, de telle sorte qu'étant donnée l'image d'un ensemble, on peut récursivement retrouver ses éléments.

**Lemme 7.** Il existe une constante  $c$  telle que, si  $u$  est une configuration, et  $A$  et  $B$ , deux sous-ensembles finis de  $\mathbb{Z}$ , et  $n = \max\{|x|, |x \in A \cup B|\}$ . On a :

$$|A \setminus B| + 2K(A) + 2K(B) + c \geq K(u/A|u/B) .$$

Si, de plus  $u$  est *log-incompressible*, on a :

$$K(u/A|u/B) \geq |A \setminus B| - 2K(A) - 2K(B) - c \log n$$

*Preuve.* Le nombre de bits présents dans  $u/A$  et pas dans  $u/B$  est  $|A \setminus B|$ . On déduit l'existence d'une constante  $c$  telle que :

$$|A \setminus B| + 2K(A) + 2K(B) + c \geq K(u/A|u/B)$$

Si, de plus,  $u$  est une configuration *log-incompressible*, on a une constante  $c_1$  telle que :

$$K(u/\llbracket -n, n \rrbracket) \geq 2n - c_1 \log n \quad (2.1)$$

D'après l'inégalité triangulaire de la complexité algorithmique (lemme 6), il existe une constante  $c_2$  telle que :

$$\begin{aligned} K(u/\llbracket -n, n \rrbracket) &\leq K(u/\llbracket -n, n \rrbracket | u/B) + K(u/B) + 2K(K(u/B)) + c_2 \\ &\leq K(u/\llbracket -n, n \rrbracket | u/B) + |B| + 2 \log |B| + c_2 \end{aligned}$$

d'où, en appliquant l'équation (2.1),

$$K(u/\llbracket -n, n \rrbracket | u/B) \geq 2n - c_1 \log n - |B| - 2 \log |B| - c_2 \quad (2.2)$$

On a aussi l'existence d'une constante  $c_3$  telle que :

$$\begin{aligned} K(u/\llbracket -n, n \rrbracket | u/B) &\leq K(u/A | u/B) + K(u/\llbracket -n, n \rrbracket \setminus A \setminus B | u/B) \\ &\quad + 2K(K(u/\llbracket -n, n \rrbracket \setminus A \setminus B | u/B)) \\ &\quad + 2K(A) + 2K(B) + c_3 \end{aligned}$$

car, pour construire les bits de  $u$  de  $\llbracket -n, n \rrbracket$ , en fonction des bits de  $u$  de  $B$ , il suffit de pouvoir construire les bits de  $u$  de  $A$  et ceux de  $\llbracket -n, n \rrbracket \setminus A \setminus B$  en fonction de ceux de  $B$ , et de connaître les ensembles  $A$  et  $B$ . On en déduit que :

$$\begin{aligned} K(u/A | u/B) &\geq K(u/\llbracket -n, n \rrbracket | u/B) - |\llbracket -n, n \rrbracket \setminus A \setminus B| \\ &\quad - 2 \log |\llbracket -n, n \rrbracket \setminus A \setminus B| - 2K(A) - 2K(B) - c_3 \end{aligned}$$

et donc, en utilisant l'équation (2.2),

$$\begin{aligned} K(u/A | u/B) &\geq 2n - c_1 \log n - |B| - 2 \log |B| - c_2 - |\llbracket -n, n \rrbracket \setminus A \setminus B| \\ &\quad - 2 \log |\llbracket -n, n \rrbracket \setminus A \setminus B| - 2K(A) - 2K(B) - c_3 \end{aligned}$$

On a  $B \cup (\llbracket -n, n \rrbracket \setminus A \setminus B) = (\llbracket -n, n \rrbracket \setminus A) \cup B$ . Comme la première union est disjointe, on en déduit que :

$$2n - |B| - |\llbracket -n, n \rrbracket \setminus A \setminus B| = 2n - |(\llbracket -n, n \rrbracket \setminus A) \cup B| = |A \setminus B| + 1$$

et donc que :

$$\begin{aligned} K(u/A | u/B) &\geq |A \setminus B| - 2K(A) - 2K(B) - 2 \log |B| - c_1 \log n \\ &\quad - 2 \log |\llbracket -n, n \rrbracket \setminus A \setminus B| - c_2 - c_3 . \end{aligned}$$

Il existe donc une constante  $c$  telle que :

$$K(u/A | u/B) \geq |A \setminus B| - 2K(A) - 2K(B) - c \log n$$

□

La même preuve, en remplaçant la définition  $x/A$  par  $x/_n A$ , donne le lemme suivant :

**Lemme 8.** *Il existe une constante  $c$  telle que, si  $u$  est une configuration, et  $A$  et  $B$ , deux sous-ensembles récurrents de  $\mathbb{Z}$ . Pour tout entier  $n$ ,*

$$K(u//_A^n | u/_n B) \leq |(A \setminus B) \cap \llbracket -n, n \rrbracket| + O(\log n) .$$

*Si, de plus,  $u$  est log-incompressible, alors cette inégalité devient une égalité.*

## 2.2 Distance Algorithmique sur les configurations bi-infinies

Comme nous l'avons déjà remarqué au chapitre 1, la notion de chaos décrite par les classes de K urka au sein de la topologie de Cantor n'est pas satisfaisante. Ainsi, nous allons d efinir une distance qui soit en rapport avec la notion de complexit e algorithmique pour tenter d'apporter une solution   ce probl eme.

Nous allons donc d efinir une distance telle que deux configurations  $x$  et  $y$  sont proches s'il est algorithmiquement facile de calculer l'une en fonction de l'autre et vice-versa. On exigera que cette distance  $d$  v erifie :

$$d(u, \sigma(u)) = 0 . \quad (2.3)$$

Dans [BGL<sup>+</sup>98], plusieurs distances algorithmiques sur les mots ont  t e introduites et  tudi es. L'un des r esultats fondamentaux de cet article est que toutes ces distances sont soit  gales   un terme logarithmique pr es, soit impropres car ne v erifient pas (2.3). Ainsi, nous n'en consid erons qu'une seule.

**D efinition 23 (pseudo-distance algorithmique).** *Soient  $x$  et  $y$ , deux configurations. La distance algorithmique  $d_{\mathcal{X}}$  entre  $x$  et  $y$  est d efinie par :*

$$d_{\mathcal{X}}(x, y) = \limsup_{n \rightarrow \infty} \frac{K(x_{-n:n}|y_{-n:n}) + K(y_{-n:n}|x_{-n:n})}{4n + 1} .$$

Telle qu'elle est d efinie,  $d_{\mathcal{X}}$  n'est qu'une pseudo-distance. Par exemple, la distance entre deux configurations qui ne diff erent qu'en une seule position est imm ediatement nulle. Soit  $\doteq$  la relation d' equivalence qui lie les configurations  $x$  et  $y$  si elles sont   distance nulle l'une de l'autre. Nous allons consid erer l'espace

$$\dot{X} = \frac{X}{\doteq}$$

qui sera notre espace de travail.

**Remarque 2.** *La d efinition de  $d$  est robuste puisqu'elle est ind ependante du mode de description optimal choisi pour la d efinition de  $K$ . Si une configuration contient  $\log n$  informations dans son pr efixe de taille  $n$ , elle est dans la m eme classe que  $\underline{0}$ . Ainsi, une classe contient un nombre ind enombrable de configurations. La classe  $\underline{0}$  inclut l'ensemble (d enombrable) des s equences r ecursives puisqu'il suffit de conna tre le programme qui la calcule, dont la taille est constante, pour obtenir n'importe quel pr efixe. Cependant, elle contient strictement plus de configurations.*

Comme nous l'avons d ej a remarqu e au chapitre pr ec edent, les propri et es topologiques d'un espace peuvent changer radicalement le comportement des syst emes dynamiques que l'on y fait  voluer (voir [BFK99]). C'est pour cette raison que nous commen ons par  tudier les propri et es topologiques de  $\dot{X}$ .

### 2.2.1 Propri et es Topologiques de l'Espace Quotient

Dans la suite, l'expression  $2^x$  signifie  $2^{2^x}$ . De plus, toutes les consid erations topologiques, comme la continuit e, la densit e ou l'ouverture, sont   consid erer dans l'espace m etrique  $(\dot{X}, d_{\mathcal{X}})$ .



**Proposition 12.** *L'espace  $\dot{X}$  n'est pas séparable, c'est-à-dire qu'il n'existe pas de sous-ensemble dénombrable dense de  $\dot{X}$ .*

*Preuve.* Soit  $A = \{a_i, i \in \mathbb{N}\}$  un sous-ensemble dénombrable de  $\dot{X}$ . Montrons qu'il existe une configuration à distance supérieure à  $\frac{1}{4}$  de tous les points de  $A$ . Définissons la suite  $(b^n)_{n \in \mathbb{N}}$  ainsi :

$$\forall k, n \in \mathbb{N}, b^{2^{k+1}n+2^k-1} = a^k .$$

Chaque  $a_i$  apparaît une infinité de fois dans la suite  $b$ .

Pour  $i > 0$ , soit  $\alpha^i = b^i_{2^{i-1}+1:2^i}$ , et soit  $\beta^i$ , un mot de longueur  $2^i - 2^{i-1}$  tel que :

$$K(\beta^i | \alpha^i) \geq 2^i - 2^{i-1}$$

Soit  $\omega$  la configuration :

$$\omega = \dots 0000000000 \beta^1 \dots \beta^{n-1} \beta^n \dots$$

où  $\beta^1$  est à la position 2 de  $\omega$ . Ainsi,

$$\omega_{2^{i-1}+1:2^i} = \beta^i$$

Montrons que, pour tout entier  $k$ ,  $d(a^k, \omega) \geq \frac{1}{4}$ . Pour tout  $n \in \mathbb{N}$ , soit  $u_n = 2^{k+1}n + 2^k - 1$ . On a :

$$\omega_{2^{u_n-1}+1:2^{u_n}} = \beta^{u_n} .$$

Or, par définition de  $b$  et  $\alpha$ , on sait que :

$$a^k_{2^{u_n-1}+1:2^{u_n}} = b^{u_n}_{2^{u_n-1}+1:2^{u_n}} = \alpha^{u_n} ,$$

d'où :

$$\begin{aligned} K(\omega_{-2^{u_n}:2^{u_n}} | a^k_{-2^{u_n}:2^{u_n}}) &\geq K(\omega_{2^{u_n-1}+1:2^{u_n}} | a^k_{2^{u_n-1}+1:2^{u_n}}) + o(2^{u_n}) \\ &\geq K(\beta^{u_n} | \alpha^{u_n}) + o(2^{u_n}) \\ &\geq 2^{u_n} - 2^{u_n-1} + o(2^{u_n}) \end{aligned}$$

et ainsi,

$$\begin{aligned} d(a^k, \omega) &\geq \limsup_{n \rightarrow \infty} \frac{K(\omega_{-2^{u_n}:2^{u_n}} | a^k_{-2^{u_n}:2^{u_n}})}{4 \cdot 2^{u_n} - 1} \\ &\geq \limsup_{n \rightarrow \infty} \frac{2^{u_n} - 2^{u_n-1} + o(2^{u_n})}{4 \cdot 2^{u_n} - 1} \\ &\geq \frac{1}{4} \end{aligned}$$

□

**Proposition 13.** *L'espace  $\dot{X}$  n'est pas localement compact.*

*Preuve.* Soit  $u$ , une configuration *log*-incompressible. Soit  $k$  un entier strictement positif. Construisons une suite de configurations  $(c^n)_{n \in \mathbb{N}}$ , dont tous les termes se trouvent dans la boule ouverte de centre  $\underline{0}$  et de rayon  $\frac{2}{k}$ , mais telle que pour tous  $q$  et  $p$ ,  $d(c^p, c^q) \geq \frac{1}{k}$ .

Soit  $k \in \mathbb{N}^*$  fixé. Pour  $n \in \mathbb{N}$ , soit  $c^n$  la configuration définie ainsi :

$$\forall i \in \mathbb{Z}, c_i^n = \begin{cases} u_{2^n(2\frac{i}{k}+1)} & \text{si } k \text{ divise } i \\ 0 & \text{sinon} \end{cases}$$

Seulement un bit tous les  $k$  peut être différent de 0. Ainsi, on a immédiatement :

$$K(c^n \text{ }_{-i:i} | 0^{2i+1}) \leq \left\lceil \frac{2i+1}{k} \right\rceil$$

et donc  $d_{\mathcal{X}}(c^n, \underline{0}) \leq \frac{1}{k}$ .

Pour tous entiers  $n, p$  et  $q$  distincts, grâce au lemme 7 utilisé avec

$$A = \{2^p(2i+1), i \in \llbracket -\lfloor \frac{n}{k} \rfloor, \lfloor \frac{n}{k} \rfloor \rrbracket\}$$

et

$$B = \{2^q(2i+1), i \in \llbracket -\lfloor \frac{n}{k} \rfloor, \lfloor \frac{n}{k} \rfloor \rrbracket\},$$

on déduit qu'il existe une constante  $c$  telle que

$$K(c^p \text{ }_{-n:n} | c^q \text{ }_{-n:n}) \geq 2 \left\lfloor \frac{n}{k} \right\rfloor + c \log n$$

car  $K(A)$  et  $K(B)$  sont calculables à l'aide de  $n$ , donc de complexité moindre que  $\log n$  à constante près. Ainsi,

$$d(c^p, c^q) \geq \limsup_{n \rightarrow \infty} \frac{2 \lfloor \frac{n}{k} \rfloor + c \log n}{2n+1} = \frac{1}{k}.$$

□

**Proposition 14.** *L'espace  $\dot{X}$  est parfait, connexe par arc et de dimension topologique infinie.*

*Preuve.* Dans cette preuve, on assimile, pour simplifier, une classe avec l'un de ses représentants.

Quel que soit le réel  $\alpha \in [0, 1]$ , soit  $(p_i^\alpha)_{i \in \mathbb{N}}$  la suite d'entiers définie par :

$$p_i^\alpha = \lfloor \alpha(2^{i+1} - 2^i) \rfloor.$$

Notons que pour  $\alpha < \beta$ ,

$$\forall i \in \mathbb{N}, p_i^\alpha < p_i^\beta.$$

Soit  $u$ , une configuration quelconque. En partant de la configuration  $u$ , on efface continûment les bits de  $u$ , jusqu'à atteindre la configuration  $\underline{0}$ . On définit la fonction  $f: \mathbb{R} \rightarrow \dot{X}$  par :

$$\forall i \in \mathbb{Z}, f(\alpha)_i = \begin{cases} u_i & \text{s'il existe } n \text{ tel que } 2^n < |i| \leq 2^n + p_n^\alpha; \\ 0 & \text{sinon} \end{cases}$$

Soit  $A^\alpha$  l'ensemble :

$$\bigcup_{n=0}^{\infty} \llbracket 2^n, 2^n + p_n^\alpha \rrbracket \cup \bigcup_{n=0}^{\infty} \llbracket -2^n, -2^n - p_n^\alpha \rrbracket \cup$$

c'est à dire l'ensemble des positions de  $f(\alpha)$  auxquelles on a recopié  $u$ . Ces ensembles sont calculables. On a donc, par définition

$$f(\alpha)_{-n:n} = u \parallel_{A^\alpha}^n$$

Ainsi, pour  $\alpha < \beta$  deux réels, grâce au lemme 8, on a donc :

$$\begin{aligned} d(f(\alpha), f(\beta)) &= \limsup_{n \rightarrow \infty} \frac{K(f(\alpha)_{-n:n} | f(\beta)_{-n:n}) + K(f(\beta)_{-n:n} | f(\alpha)_{-n:n})}{4n + 2} \\ &\leq \limsup_{n \rightarrow \infty} \frac{0 + |\llbracket -n, n \rrbracket \cap (A^\beta \setminus A^\alpha)|}{4n + 2} \end{aligned}$$

Si, de plus  $u$  est *log*-incompressible, on a :

$$d(f(\alpha), f(\beta)) = \limsup_{n \rightarrow \infty} \frac{|\llbracket -n, n \rrbracket \cap (A^\beta \setminus A^\alpha)|}{4n + 2}$$

Si on note  $\gamma(n)$  l'entier tel que  $2^{\gamma(n)} < n \leq 2^{\gamma(n+1)}$ , on a :

$$|A^\beta \setminus A^\alpha \cap \llbracket -n, n \rrbracket| = 2 \left( \sum_{i=0}^{\gamma(n)-1} (p_i^\beta - p_i^\alpha) + \max(\min(p_{\gamma(n)}^\beta, n) - p_{\gamma(n)}^\alpha, 0) \right) .$$

Par définition de la limite supérieure, on peut ne considérer que les  $n$  correspondant à tous les maxima locaux.

$$\limsup_{n \rightarrow \infty} \frac{|\llbracket -n, n \rrbracket \cap (A^\beta \setminus A^\alpha)|}{4n + 2} = \limsup_{n \rightarrow \infty} \frac{2 \sum_{i=0}^n p_i^\beta - p_i^\alpha}{4(2^n + p_n^\beta)}$$

Comme  $\sum_{i=0}^n 2^i$  est négligeable devant  $2^{n+1}$ , on déduit que :

$$d(f(\alpha), f(\beta)) \leq \limsup_{n \rightarrow \infty} \frac{p_n^\beta - p_n^\alpha}{2(2^n + p_n^\beta)} \quad (2.4)$$

qui est une égalité si  $u$  est *log*-incompressible.

*Continuité de  $f$ .* Soient  $\alpha < \beta$  deux réels. On a, d'après l'équation précédente :

$$\begin{aligned} d(f(\alpha), f(\beta)) &\leq \limsup_{n \rightarrow \infty} \frac{p_n^\beta - p_n^\alpha}{2(2^n + p_n^\beta)} \\ &\leq \limsup_{n \rightarrow \infty} \frac{p_n^\beta - p_n^\alpha}{2(2^n - 2^{n-1})} \\ &= \frac{1}{2}(\beta - \alpha) . \end{aligned}$$

On en déduit que  $f$  est  $\frac{1}{2}$ -lipschitzienne et donc continue. Par conséquent, on a construit pour toute configuration  $u$ , une fonction continue de  $[0, 1]$  dans  $\dot{X}$  tel que  $f(0) = \underline{0}$ ,  $f(1) = u$ . On en déduit que  $\dot{X}$  est connexe par arc.

*Perfection.* Comme la séquence  $f(\frac{1}{n})$  a pour limite  $u$ , on en déduit que  $u$  est un point d'accumulation. Il n'y a donc pas de points isolés.

Dans la suite, on suppose que  $u$  est *log*-incompressible.

*Injectivité de  $f$ .* Soient  $\alpha < \beta$  deux réels. Grâce à l'équation (2.4), on a, comme  $p_n^\beta \leq 2^n - 2^{n-1}$ , et que  $2^{n-1}$  est négligeable devant  $2^n$ ,

$$\begin{aligned} d(f(\alpha), f(\beta)) &= \limsup_{n \rightarrow \infty} \frac{p_n^\beta - p_n^\alpha}{2^n - 2^{n-1}} \\ &\geq \beta - \alpha . \end{aligned}$$

La fonction  $f$  est donc injective.

*Dimension infinie de  $\dot{X}$ .* On notera  $f_u$  la fonction  $f$  associée à la configuration  $u$ . La continuité et l'injectivité de  $f_u$  montrent que  $\dot{X}$  est au moins de dimension 1 puisqu'elle plonge  $[0, 1]$  dans cet espace. Soit  $n$  un entier positif. Soit  $h$  la fonction définie ainsi :

$$h : \begin{cases} (\dot{X})^n & \rightarrow \dot{X} \\ (c^i)_{1 \leq i \leq n} & \mapsto h((c^i)_{1 \leq i \leq n}) \end{cases}$$

avec

$$h((c^i)_{1 \leq i \leq n}) = \dots c_{-k}^1 c_{-k}^2 \dots c_{-k}^n c_{-k+1}^1 \dots c_{-k+1}^n \dots c_k^1 c_k^2 \dots c_k^n \dots .$$

La fonction  $h$  est continue :

$$\forall x, y \in (\dot{X})^n, \forall \delta \in \mathbb{R} \forall i, d(x^i, y^i) \leq \delta \implies d(h(x), h(y)) \leq \delta$$

car :

$$K(h(x)_{-k:k} | h(y)_{-k:k}) \leq \sum_{i=1}^n K(x^i_{-\lceil \frac{k}{n} \rceil : \lceil \frac{k}{n} \rceil} | y^i_{-\lceil \frac{k}{n} \rceil : \lceil \frac{k}{n} \rceil}) + n \log \left\lceil \frac{k}{n} \right\rceil .$$

Soient  $u^1, \dots, u^n$ , suite de  $n$  configurations *log*-incompressibles, et *log*-indépendantes, c'est-à-dire qu'il existe une constante  $c$  telle que pour tout  $i, \neq j$  entre 1 et  $n$ , pour tout  $k$ ,

$$K(u^i_{-k:k} | u^j_{-k:k}) \geq 2k - c \log k .$$

Par exemple, si  $u$  est une configuration *log*-incompressible, alors les configurations

$$u^i = u /_{nZ+i-1}$$

vérifient, grâce au lemme 7, la condition ci-dessus.

Soit  $g_n$  définie par :

$$g_n : \begin{cases} [0, 1]^n & \rightarrow \dot{X} \\ (\alpha^i)_{1 \leq i \leq n} & \mapsto h(f_{u^1}(\alpha_1), f_{u^2}(\alpha_2), \dots, f_{u^n}(\alpha_n)) . \end{cases}$$

Comme  $g_n$  est la composition  $f$  et  $h$ , elle est continue. Montrons qu'elle est injective. Soient  $\alpha_1, \dots, \alpha_n$ ,  $n$  réels,  $i$  un entier de  $\llbracket 1, i \rrbracket$ , et  $\beta$  un réel strictement supérieur à  $\alpha_i$ . Soient :

$$\begin{aligned} x &= g_n(\alpha_1, \dots, \alpha_i, \dots, \alpha_n) \text{ et} \\ y &= g_n(\alpha_1, \dots, \beta, \dots, \alpha_n) . \end{aligned}$$

On a, grâce au lemme 7, au fait que les  $u^i$  sont *log*-incompressibles et *log*-indépendantes, et grâce à l'équation (2.4),

$$\begin{aligned}
d(x, y) &\geq \limsup_{k \rightarrow \infty} \frac{K(y_{-k:k} | x_{-k:k})}{4k + 1} \\
&\geq \limsup_{k \rightarrow \infty} \frac{K(y_{-k:k} | x_{-k:k})}{4k + 1} \\
&\geq \limsup_{k \rightarrow \infty} \frac{K\left(y / \lfloor \frac{k}{n} \rfloor \mid x / \lfloor \frac{k}{n} \rfloor\right)}{4k + 1} \\
&\geq \limsup_{k \rightarrow \infty} \frac{K\left(f_{u^i}(\beta)_{-\lfloor \frac{k}{n} \rfloor : \lfloor \frac{k}{n} \rfloor} \mid f_{u^i}(\alpha_i)_{-\lfloor \frac{k}{n} \rfloor : \lfloor \frac{k}{n} \rfloor}\right)}{4k + 1} \\
&\geq \beta - \alpha_i
\end{aligned}$$

Pour tout entier  $\mathbb{N}$ , la fonction  $g_n$  est continue et injective, et plonge  $[0, 1]^n$  dans  $\dot{X}$ . L'espace  $\dot{X}$  est donc de dimension topologique infinie.  $\square$

Là encore, il nous manque la complétude de l'espace  $\dot{X}$ .

**Problème Ouvert 2.** *L'espace  $\dot{X}$  est-il complet ?*

## 2.2.2 Automates Cellulaires et Continuité de l'Information

Dans cette section, nous allons utiliser deux notions.

**Définition 24 (compatibilité avec  $\dot{=}$ ).** *Un automate cellulaire est compatible avec  $\dot{=}$  si et seulement si, pour toutes configurations  $x$  et  $y$ ,*

$$x \dot{=} y \implies f(x) \dot{=} f(y) .$$

*Si un automate cellulaire  $f$  est compatible avec  $\dot{=}$ , on note  $\dot{f}$  la fonction qui à toute classe  $x$  associe la classe contenant les images des configurations de  $x$ .*

Dans ce chapitre, on dira simplement compatible pour signifier compatible avec  $\dot{=}$ .

Cependant, même si une fonction n'est pas compatible, on peut quand même parler de fonction continue si elle vérifie la propriété

$$\forall x \in X, \forall \varepsilon > 0, \exists \delta, \forall y, d_X(x, y) < \delta \implies d_X(f(x), f(y)) < \varepsilon .$$

**Définition 25 (automate cellulaire pré-injectif).**

*Un automate cellulaire est pré-injectif si la restriction de sa règle globale aux configurations finies est injective.*

Myhill et Moore ont démontré dans [Moo63, Myh63] le théorème suivant :

**Théorème 5 (Théorème de Myhill et Moore).**

Un automate cellulaire  $f$  de règle locale  $\lambda$ , et dont l'état 0 vérifie

$$\lambda(0, 0, \dots, 0) = 0$$

est surjectif si et seulement s'il est pré-injectif.

**Lemme 9.** Si un automate cellulaire  $f$  est surjectif, alors il est compatible et  $\hat{f}$  est l'identité sur  $\hat{X}$ .

*Preuve.* Soit  $f$  un automate cellulaire surjectif de rayon  $r$ .

Dans [MK76], il est prouvé que, comme  $f$  est surjectif,  $f$  est balancé, c'est-à-dire que tous les mots de longueur  $k$  ont le même nombre d'antécédents (de taille  $k + 2r$ ) par  $f$ . Comme le nombre de mots de longueur  $k$  est  $2^k$ , il y a exactement  $2^{2r}$  antécédents d'un mot donné de longueur  $k$ .

- Soit  $p$  un programme qui sur l'entrée  $\langle y, i \rangle$  effectue les opérations suivantes :
- crée l'ensemble  $P$  des  $2^{2r}$  antécédents de  $y$ ;
  - renvoie le  $i^{\text{ème}}$  mot de  $P$ , dans l'ordre alphabétique, tronqué à ses  $|y|$  lettres centrales

quand  $i \leq 2^{2r}$ , et qui ne s'arrête pas sinon.

Ainsi, la seule chose nécessaire au calcul de  $x_{-n:n}$  à partir de  $f(x)_{-n:n}$  dans le mode de description  $p$  est le numéro d'ordre de  $x_{-n:n}$  dans l'ensemble  $P$  des antécédents de  $f(x)_{-n:n}$ , dont le cardinal est borné. On en déduit que :

$$\forall n \in \mathbb{N}, K(x_{-n:n} | f(x)_{-n:n}) = O(1) .$$

Comme il suffit de connaître  $f$  et les  $2r$  bits manquants pour calculer le mot  $f(x)_{-n:n}$  à partir de  $x_{-n:n}$ , on a  $d(x, f(x)) = 0$ . Par conséquent,  $f$  est compatible et  $\hat{f}$  est l'identité sur  $\hat{X}$ .  $\square$

Comme l'identité est continue, on déduit que :

**Corollaire 3.** Les automates cellulaires surjectifs sont continus.

Le résultat précédent peut être renforcé de la manière suivante :

**Proposition 15.** Les automates cellulaires continus sont ou bien surjectifs ou bien constants. Les automates cellulaires non continus ne sont pas compatibles.

*Preuve.* Soit  $g$  un automate cellulaire ni surjectif, ni constant, de rayon  $r$ . Soit  $s$  l'état obtenu en appliquant la règle locale à  $0^{2r+1}$ . On définit  $h$ , une fonction lettre à lettre sur les configurations qui associent 0 à  $s$ ,  $s$  à 0 et les autres états sur eux-mêmes. Soit  $f = h \circ g$ . Comme  $h$  est une involution,  $f$  est aussi un automate cellulaire ni surjectif, ni constant.

Grâce au théorème de Myhill et Moore, la restriction de  $f$  aux configurations finies n'est pas pré-injectif, et donc il existe deux configurations finies telles que :

$$0^\infty w_0 0^\infty \rightarrow 0^\infty z_0 0^\infty \quad \text{et} \quad 0^\infty w_2 0^\infty \rightarrow 0^\infty z_0 0^\infty ,$$

pour des mots  $w_0, w_2, z_0$  convenables. De plus, comme  $f$  n'est pas constant, il existe au moins une configuration  $0^\infty w_1 0^\infty$  dont l'image est  $0^\infty z_1 0^\infty$ , avec  $z_1 \neq z_0$ . Sans perte de généralité, on peut supposer que  $|w_0| = |w_1| = |w_2|$ , et

que  $|z_1| = |z_0| = |w_0| + 2r$ . Soit  $x$  une configuration *log*-incompressible constituée de 0 et 1. Soient  $\alpha$  et  $\beta$ , deux configurations telles que :

$$\alpha = \dots 0^r w_{x_i} 0^r w_{x_{i+1}} 0^r \dots$$

et

$$\beta = \dots 0^r w_{2x_i} 0^r w_{2x_{i+1}} 0^r \dots$$

On sait que  $\alpha \doteq \beta$ , et que :

$$f(\alpha) = \dots z_{x_i} z_{x_{i+1}} \dots \quad \text{et} \quad f(\beta) = z_0^\infty$$

et que  $d(f(\alpha), f(\beta)) > 0$ .

Comme  $g = h \circ f$ , et que  $h$  est juste un renommage des états,  $d(g(\alpha), g(\beta)) > 0$ . Ainsi,  $g$  n'est pas compatible, et par conséquent, non continue.

Réciproquement, si  $g$  est constante, elle est continue, et avec le lemme 3, si  $g$  est surjective, elle est continue.  $\square$

La proposition précédente caractérise précisément l'ensemble des automates cellulaires continus sur  $\dot{X}$ . En dimension un, la classe des automates cellulaires continus est décidable puisque la surjectivité est décidable[AP72, Sut91].

Bien que les automates cellulaires non surjectifs ne soient pas continus, le résultat suivant montre qu'il reste cependant un peu de continuité.

**Proposition 16.** *Pour tout automate cellulaire  $f$ ,*

- i) La classe  $\underline{0}$  est un point fixe de  $f$  dans  $\dot{X}$  ;*
- ii) La classe  $\underline{0}$  est un point d'équicontinuité de  $f$ .*

*Preuve.* i) Toute configuration  $x$  de la classe de  $\underline{0}$  vérifie la propriété suivante,

$$\limsup_{n \rightarrow \infty} \frac{K(x_{-n:n}|n)}{n} = 0 .$$

Par inégalité triangulaire (lemme 6), on a, pour tout entier  $n$ ,

$$K(f(x)_{-n:n}|n) \leq K(f(x)_{-n:n}|x_{-n:n}) + K(x_{-n:n}|n) + O(1)$$

et comme  $K(f(x)_{-n:n}|x_{-n:n}) = O(1)$ ,

$$\limsup_{n \rightarrow \infty} \frac{K(f(x)_{-n:n}|n)}{n} = 0 .$$

La configuration  $f(x)$  est donc dans la même classe que  $\underline{0}$ .

- ii) Comme  $K(f^t(x)_{-n:n}|x_{-n:n}) = O(1)$  puisque la connaissance de la règle locale de l'automate cellulaire et des  $2rt$  bits manquants sont suffisants pour calculer  $f^t(x)_{-n:n}$  à partir de  $x_{-n:n}$ , on a :

$$K(f^t(x)_{-n:n}|n) \leq K(x_{-n:n}|n) + O(1)$$

et donc :

$$d(f^t(x), \underline{0}) \leq d(x, \underline{0})$$

ce qui prouve que le point  $\underline{0}$  est d'équicontinuité.  $\square$

Il n'est pas toujours intéressant d'étudier le comportement de l'automate cellulaire sur l'ensemble entier des configurations. Par exemple, les configurations qui ne sont pas dans l'ensemble image ne sont pas caractéristiques pour l'étude de l'évolution de l'automate. C'est pour cette raison que la notion d'ensemble limite a été introduite.

**Définition 26 (sous-shift).** *Un ensemble  $S \subset X$  est un sous-shift s'il est fermé pour la topologie de Cantor et invariant par décalage.*

Comme l'ensemble est fermé et invariant par décalage, le complémentaire est une réunion de cylindres invariante par décalage. On appelle *mot interdit* du sous-shift l'ensemble des générateurs de ces cylindres. Si cet ensemble est fini, on dit que le sous-shift est de type fini. Ainsi, une configuration est dans le sous-shift si et seulement si elle ne contient pas de mot interdit.

**Définition 27 (ensemble limite).** *Soit  $f$  un automate cellulaire. L'ensemble limite de cet automate est l'ensemble noté  $\Omega(f)$  défini par :*

$$\Omega(f) = \bigcap_{n \in \mathbb{N}} f^n(X)$$

L'ensemble limite est l'ensemble des configurations pour lesquelles il existe, pour tout entier  $n$ , un antécédent par  $f^n$ . On notera  $\Omega^n(f)$  le langage des mots de taille  $n$  apparaissant dans une des configurations de  $\Omega(f)$ . Cet ensemble est un *sous-shift*. En effet, il est invariant par translation car  $f$  aussi. Il est fermé puisque c'est une intersection d'ensembles fermés. En effet,  $f^n(X)$  est l'image d'un fermé par une application continue sur un compact. Ainsi, il est compact et par conséquent fermé. Il est aussi récursivement énumérable car pour semi-décider si un motif est interdit, il suffit de construire l'ensemble des ses antécédents par  $f^i$ , pour  $i \in \mathbb{N}$ . Si on ne peut en trouver, le motif est interdit.

Une question naturelle est de se demander si les automates cellulaires non surjectifs sont continus sur leur ensemble limite, où ils agissent surjectivement. La proposition suivante est une réponse partielle à cette question.

**Théorème 6.** *Soit  $g: \mathbb{N} \rightarrow \mathbb{N}$  une fonction calculable telle que  $g(n) = o(n)$ . Soit  $f$  un automate cellulaire de rayon  $r$ . Si la condition suivante, qui limite le nombre d'antécédents, est vérifiée*

$$\forall x \in X, \forall n \in \mathbb{N}, \log |f^{-1}(x_{-n:n}) \cap \Omega^{n+2r}(f)| \leq g(n) \quad (2.5)$$

*alors,  $f$  est compatible sur  $\Omega(f)$  et  $\hat{f}|_{\Omega(f)}$  est égale à l'identité de  $\Omega(f)$ .*

*Preuve.* On a toujours :

$$\limsup_{n \rightarrow \infty} \frac{K(f(x)_{-n:n} | x_{-n:n})}{n} = 0 .$$

Il reste à prouver que, pour tout  $x$  de  $\Omega(f)$ ,

$$\limsup_{n \rightarrow \infty} \frac{K(x_{-n:n} | f(x)_{-n:n})}{n} = 0 .$$

Soit  $p$  le programme, qui, sur l'entrée  $\langle y, i \rangle$ , effectue les opérations suivantes :



- crée l'ensemble  $P$  de tous les antécédents du mot  $y$ ;
- énumère l'ensemble des mots interdits de  $\Omega(f)$ , en effaçant, à chaque nouveau mot, les éléments de  $P$  qui contiennent ce mot, jusqu'à ce qu'il contienne moins de  $2^{g(|y|)}$  mots;
- renvoie le  $i^{\text{ème}}$  mot de  $P$ , selon l'ordre lexicographique.

Ce programme est correctement défini car  $F$  est récursivement énumérable. Comme la condition (2.5) est vérifiée,  $p$  s'arrête toujours. On a donc, pour  $x$  dans  $\Omega(f)$ ,

$$K(x_{-n-2r:n+2r} | f(x_{-n:n})) = g(n)$$

et donc

$$d(x, f(x)) = 0 .$$

□

Cependant, on conjecture qu'il existe une telle  $g$  pour tout automate cellulaire. Il reste donc le problème ouvert suivant :

**Problème Ouvert 3.** *Soit  $f$  un automate cellulaire de rayon  $r$ . Existe-t-il une fonction calculable  $g: \mathbb{N} \rightarrow \mathbb{N}$  telle que  $g(n) = o(n)$  et vérifiant :*

$$\forall x \in X, \forall n \in \mathbb{N}, \log |f^{-1}(x_{-n:n}) \cap \Omega^{n+2r}(f)| \leq g(n) ? \quad (2.6)$$

La proposition suivante permet de synthétiser le comportement des automates cellulaires sur  $\dot{X}$ .

**Proposition 17.** *Dans  $\dot{X}$ , les propositions suivantes sont vraies :*

- i) Les propriétés de log-incompressibilité et la faible-incompressibilité sont conservées par l'application d'un automate cellulaire surjectif;*
- ii) Les automates cellulaires non surjectifs détruisent ces deux propriétés;*
- iii) Tout automate cellulaire préserve le fait de ne pas être faiblement-compressible.*

*Preuve.* *i)* Lemme 9;

*ii)* Proposition 15;

*iii)* Proposition 16. □

## Chapitre 3

# Régularité et récursivité des Pavages

### 3.1 Généralités sur les pavages

Ce chapitre<sup>1</sup> est dédié à l'étude des pavages. La notion de complexité, aussi bien structurelle qu'algorithmique semble occuper une position centrale dans la poursuite de l'étude des pavages. En effet, après les découvertes de pavages quasipériodiques, il appert que les pavages sont bien plus complexes et expressifs que ce l'on pouvait initialement penser.

En effet, dans l'espoir de trouver un monde mathématique défini par une axiomatique fondée sur la logique du premier ordre, et qui soit complet et décidable, Hilbert avait énoncé un problème appelé *Das Entscheidungsproblem* dans lequel il est question de la décidabilité des formules dans la logique du premier ordre. C'est à la suite de la démonstration par Berger en 1966 dans [Ber66] du problème de pavabilité du plan (le *domino problem*), puis par Gurevich et Koryakov dans [GK72] que les ensembles de tuiles dont les pavages sont tous périodiques ne sont pas récursivement séparable de ceux n'admettant pas de pavages, que *Das Entscheidungsproblem* a été démontré de manière satisfaisante indécidable pour une des classes de formules logiques, la classe de Kahr  $K = [\forall\exists\forall, (0, \omega)]$ . En effet, il a été construit une réduction qui transforme un ensemble de tuiles en une formule de cette classe qui est satisfaisable si et seulement si l'ensemble pave le plan, et qui admet un modèle *fini* si et seulement si l'ensemble pave le plan *périodiquement*. À la suite de ce résultat, le problème général a été fermé, grâce à des réductions entre les classes de fonctions. La puissance des pavages dans ce domaine est telle qu'aucune preuve s'en passant n'existe.

Pour étudier le caractère plus ou moins régulier des pavages, on utilise la notion de quasipériodicité. Cette notion est une généralisation de la périodicité, cruciale comme le montre le théorème suivant : “tout ensemble de tuiles pavant le plan le pave aussi de manière quasipériodique [Dur99]” : en effet, pour tout ensemble de tuiles qui pave le plan, on peut donc exprimer sa complexité en caractérisant le degré de quasipériodicité de son pavage le plus simple. Les

---

<sup>1</sup>Ce chapitre a été l'objet de la publication [CD00]

propriétés de ces pavages quasipériodiques nous éclairent sur la façon dont l'ensemble de tuiles considéré pave le plan.

### 3.1.1 Tuiles de Wang et autres modèles de pavage

Il existe différents modèles pour représenter les pavages du plan à l'aide de tuiles placées sur une grille. Ceux que nous utiliserons sont tous équivalents. Il s'agit de tuiles colorées, de tuiles fléchées ou de formes dont les sommets sont à coordonnées rationnelles. On peut aussi mélanger les différents types de tuiles, même si la notion peut devenir plus lourde à formaliser.

Ainsi, pour paver le plan, on se donne un ensemble *fini* de tuiles. À chaque type de tuile est associé une notion de pavabilité définissant précisément quelles sont les règles à respecter pour former un pavage correct du plan à l'aide des tuiles de l'ensemble.

Dans la suite, tout ensemble de tuiles considéré est supposé être *fini*.

#### 3.1.1.1 Tuiles colorées, ou tuiles de Wang

Les tuiles *colorées*, ou *tuiles de Wang*, sont des tuiles carrées dont les côtés sont colorés. Un ensemble de tuiles est un ensemble fini de ces tuiles colorées, et pour paver le plan, on a le droit de dupliquer à volonté les tuiles de cet ensemble, mais jamais de les faire pivoter ou de leur faire subir une symétrie (sinon le plan serait toujours pavable (Fig. 3.1)).

Plus formellement :

**Définition 28 (tuiles colorées).** Soit  $\mathcal{C}$  un ensemble fini, appelé ensemble de couleurs. Une tuile colorée est un quadruplet  $(h, b, g, d)$  d'éléments de  $\mathcal{C}$ .

La première coordonnée du quadruplet est la couleur située en haut de la tuile, la seconde en bas, la troisième à gauche et la dernière à droite (Fig. 3.2).

Ainsi, on obtient la notion de pavabilité suivante :

**Définition 29.** Soit  $\tau$  un ensemble de tuiles colorées. Un pavage du plan par  $\tau$  est une application  $P$  de  $\mathbb{Z}^2$  dans  $\tau$  vérifiant :

$$\forall x \in \mathbb{Z}^2, \begin{cases} \pi_4(x) = \pi_3(x + (0, 1)) \\ \pi_2(x) = \pi_1(x + (1, 0)) \end{cases}$$

où, pour  $i \in \{1, 2, 3, 4\}$ ,  $\pi_i(a)$  est la  $i^{\text{ème}}$  coordonnée de  $a$ .

#### 3.1.1.2 Tuiles fléchées

Les tuiles fléchées sont une variante des tuiles colorées. Cette fois-ci, au lieu de mettre des couleurs sur les côtés des tuiles, on y place des flèches, de différentes couleurs ou formes (doubles, triples, etc.) allant vers le milieu de la tuile ou vers l'extérieur de la tuile. Pour fabriquer un pavage correct, on impose à l'une des flèches du côté commun de deux tuiles adjacentes d'être le prolongement de l'autre (Fig. 3.3).

Formellement :

**Définition 30 (tuiles fléchées).** Soit  $\mathcal{F}$  un ensemble fini, appelé ensemble de types de flèches. L'ensemble des flèches possibles est l'ensemble  $\mathcal{F} \times \{E, S\}$  (ou  $E$  signifie flèche entrante, et  $S$  flèche sortante). Une tuile fléchée est un quadruplet  $(h, b, g, d)$  d'éléments de  $\mathcal{F} \times \{E, S\}$ .

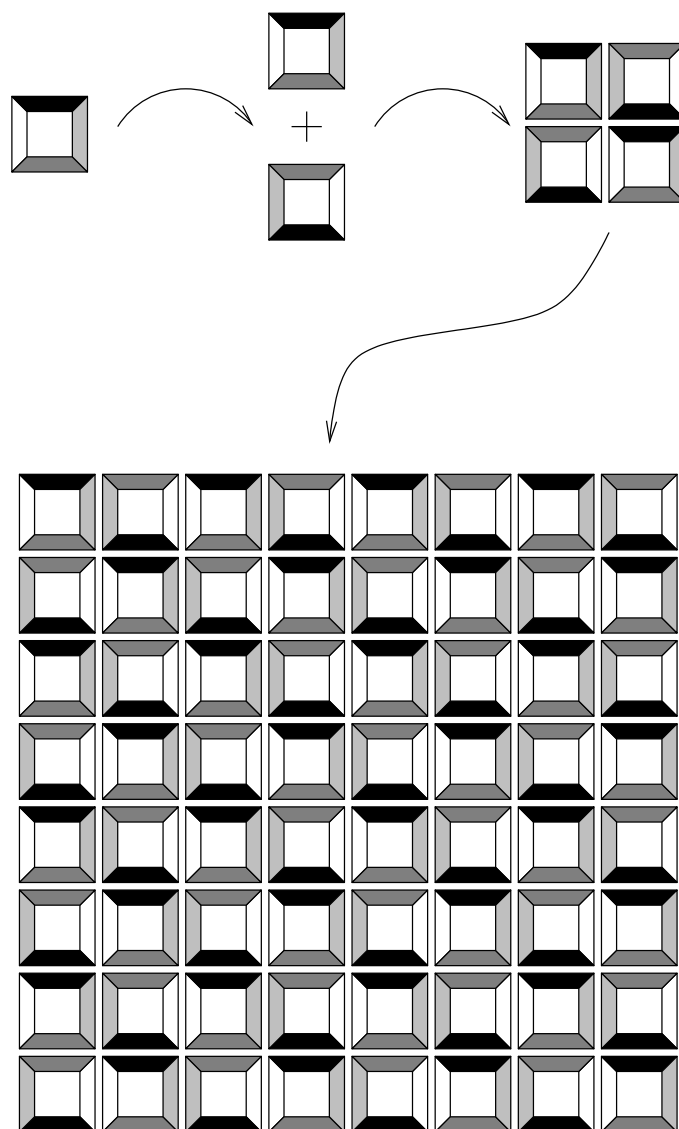


FIG. 3.1 – En autorisant les demi-tours, tout ensemble de tuiles pave le plan périodiquement

$$(h, b, g, d) = \begin{array}{|c|} \hline h & d \\ \hline g & b \\ \hline \end{array}$$

FIG. 3.2 – Exemple de tuile colorée

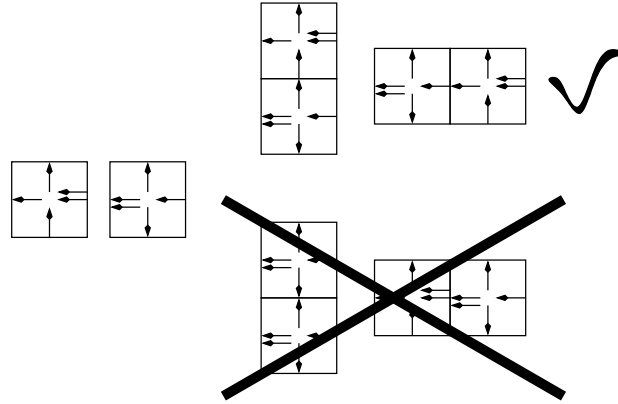


FIG. 3.3 – Tuiles fléchées : à gauche, les tuiles ; en haut à droite, des juxtapositions autorisées ; en bas à droite, d'autres, interdites.

Comme pour les tuiles colorées, la première coordonnée du quadruplet est la flèche située en haut de la tuile, la seconde en bas, la troisième à gauche et la dernière à droite.

Ainsi, on obtient la notion de pavabilité suivante :

**Définition 31.** Soit  $\tau$  un ensemble de tuiles fléchées. Un pavage du plan par  $\tau$  est une application  $P$  de  $\mathbb{Z}^2$  dans  $\tau$  vérifiant :

$$\forall x \in \mathbb{Z}^2, \begin{cases} \pi_4(x) \sim \pi_3(x + (0, 1)) \\ \pi_2(x) \sim \pi_1(x + (1, 0)) \end{cases}$$

où, pour  $i \in \{1, 2, 3, 4\}$ ,  $\pi_i(a)$  est la  $i^{\text{ème}}$  coordonnée de  $a$ , et  $(t, f) \sim (t', f')$  si et seulement si  $t = t'$  et  $f \neq f'$ .

### 3.1.1.3 Formes dont les sommets sont à coordonnées rationnelles

Les formes dont les sommets sont à coordonnées rationnelles sont les pavages les plus naturels<sup>2</sup> : chaque pièce est un polygone non croisé (mais pas spécialement convexe), et l'on pave le plan en juxtaposant les pièces de manière à ne laisser aucun trou.

Formellement :

**Définition 32 (Formes rationnelles).** Une forme dont les sommets sont à coordonnées rationnelles, ou plus simplement forme rationnelle, est un polygone non croisé de  $\mathbb{R}^2$  dont les sommets sont à coordonnées rationnelles.

Pour cette notion d'ensemble de tuiles, la définition intuitive de pavabilité peut être formalisée de la manière suivante :

<sup>2</sup>Les pavages de Penrose ne rentrent pas dans cette catégorie car les coordonnées des sommets ne sont pas rationnelles[GS86]

**Définition 33.** Soit  $\tau$  un ensemble de formes rationnelles. Un pavage du plan à l'aide de ces formes est un ensemble de couples  $(i, \tau_i) \in \mathbb{Q}^2 \times \tau$  tel que :

$$\overline{\bigcup_{(i, \tau_i) \in \tau} i + \tau_i} = \mathbb{R}^2$$

$$\forall (i, \tau_i), (j, \tau_j) \in \tau, \overset{\circ}{(i + \tau_i) \cap (j + \tau_j)} = \emptyset,$$

où  $\overline{\quad}$  signifie "adhérence de" et  $\overset{\circ}{\quad}$  signifie "intérieur de".

La première formule assure que le pavage couvre bien tout le plan, la seconde que jamais deux pièces ne se recouvrent.

### 3.1.1.4 Contraintes locales

Les pavages par contraintes locales ne ressemblent pas aux types de pavages précédents. Cette fois-ci, un pavage est une assignation à une lettre de chaque case du domaine à paver, de telle sorte qu'en tout point du domaine une *contrainte locale* soit vérifiée. Dans cette thèse, on considère les pavages formés de seulement deux lettres que l'on notera 0 et 1, que l'on nommera *configuration binaire*.

**Définition 34 (contrainte locale).** On appelle voisinage un vecteur (ou suite finie) d'éléments distincts de  $\mathbb{Z}^2$ . Pour  $V = (v_1, \dots, v_n)$ , on appelle contrainte  $V$ -locale une fonction de  $\{0, 1\}^n$  dans  $\{0, 1\}$ . On appelle contrainte locale un couple  $(V, f)$  dans lequel  $f$  est une contrainte  $V$ -locale. La fonction  $f$  est appelée la fonction de contrainte de la contrainte locale.

**Définition 35 (configuration binaire).** Une configuration binaire est une fonction de  $\mathbb{Z}^2$  dans  $\{0, 1\}$ .

Dans ce cas, la notion de pavabilité devient :

**Définition 36.** Soit  $f$  une contrainte locale. Une configuration binaire  $p$  est produite par, ou vérifie, la contrainte locale  $((v_1, \dots, v_n), f)$  si, pour tout  $x$  de  $\mathbb{Z}^2$  :

$$f(p(x + v_1), p(x + v_2), \dots, p(x + v_n)) = 0.$$

## 3.1.2 Résultats sur les pavages

### 3.1.2.1 Procédé d'extraction diagonale, lemme de König

Afin de démontrer qu'un ensemble de tuiles peut paver le plan tout entier, on utilise souvent un *procédé d'extraction diagonale*. Ce procédé permet de démontrer le théorème suivant :

**Lemme 10 (lemme de König).** Si on peut paver des motifs carrés de plus en plus grands à l'aide d'un ensemble de tuiles, alors cet ensemble de tuiles pave le plan.

*Preuve.* Soit  $\mathcal{C}$  l'ensemble des motifs carrés pavés correctement. Quitte à supprimer une ligne et une colonne, on supposera que chaque motif est de côté

impair, et que par conséquent, la notion de tuile centrale, ou de sous-motifs centraux, est toujours bien définie. Nous allons fabriquer une suite croissante de motifs pavés correctement, c'est-à-dire telle que tout motif de la suite est un motif central de tous les motifs qui le suivent.

Considérons tout d'abord les tuiles centrales de tous les motifs de  $\mathcal{C}$ . Comme l'ensemble des tuiles est fini, il y en a un nombre fini. Par conséquent, il y a un nombre infini de motifs ayant la même tuile centrale. Soit  $\mathcal{C}_0$  cet ensemble. Le motif carré de côté 1 uniquement constitué de cette tuile centrale est le premier élément de notre suite croissante. On le note  $M_0$ .

La suite du procédé est récursive. Voici comment on passe d'une étape à la suivante, c'est-à-dire du couple  $(\mathcal{C}_n, M_n)$  à  $(\mathcal{C}_{n+1}, M_{n+1})$ . Considérons l'ensemble des motifs centraux de taille  $2n + 1$  des motifs de  $\mathcal{C}_n$  auxquels on retire les motifs trop petits. Il y a un nombre fini de motifs possibles, à savoir  $t^{(2n+1)^2}$  si  $t$  est le nombre de tuiles possibles. Par conséquent, il y a un nombre infini de motifs de  $\mathcal{C}_n$  ayant le même motif central de taille  $2n + 1$ . Soit  $\mathcal{C}_{n+1}$  cet ensemble, et  $M_{n+1}$  ce motif.

Il est aisé de voir que la suite  $M$  est bien une suite croissante, car en effet, pour tout entier  $n$ , le motif  $M_{n+1}$  est un motif central de motifs ayant tous  $M_n$  comme motif central.

Ainsi, on peut en déduire un pavage complet du plan : la tuile placée à la case  $(i, j)$  sera celle qui se trouve à la case  $(i, j)$  d'un motif de la suite défini sur cette case. Le résultat obtenu est bien un pavage puisque les motifs sont correctement pavés.  $\square$

Cette notion correspond à la notion de compacité de l'ensemble des pavages. La manière d'obtenir une suite croissante de motifs correspond en fait à l'extraction d'une sous-suite convergente.

### 3.1.2.2 Rapports entre les différentes définitions

La robustesse de la notion de pavage est due au résultat suivant :

**Théorème 7.** *Les quatre notions de pavages introduites dans la section précédente sont toutes équivalentes, c'est-à-dire qu'il existe des manières de coder une représentation à l'aide d'une autre qui conserve les propriétés de pavabilité. Ainsi, pour deux types de tuiles  $\mathcal{T}_1$  et  $\mathcal{T}_2$ , il existe une transformation récursive  $f$  telle que, le plan est pavable grâce aux tuiles de type  $\mathcal{T}_1$  d'un ensemble  $T$  si et seulement si le plan est pavable grâce aux tuiles de type  $\mathcal{T}_2$  de l'ensemble  $f(T)$ .*

*Preuve.* On ne donnera pas de preuve formelle complète, car elle est très technique. On se contentera de décrire les transformations, et de justifier pourquoi elles conviennent.

Il est simple de traduire des tuiles colorées en tuiles fléchées. Il suffit de remarquer qu'une couleur située sur la face supérieure d'une tuile ne peut être juxtaposée qu'à la même couleur située sur la face inférieure d'une autre. Ainsi la même couleur, mais sur des faces opposées, jouent des rôles différents. Si  $\tau$  est un ensemble de tuiles colorées, et  $\mathcal{C}$  l'ensemble des couleurs, soit  $\tau'$ , l'ensemble de tuiles fléchées dont l'ensemble des types de flèches est  $\mathcal{C}$ , et dont les tuiles sont :

$$\{((h, E), (b, S), (g, E), (d, S)), (h, b, g, d) \in \tau\} .$$

Ces transformations vérifient le théorème. En effet, si un domaine est pavable à l'aide de tuiles de  $\tau$ , il est pavable à l'aide de tuiles de  $\tau'$  en remplaçant toute tuile  $(h, b, g, d)$  par  $((h, E), (b, S), (g, E), (d, S))$  (Fig. 3.4).

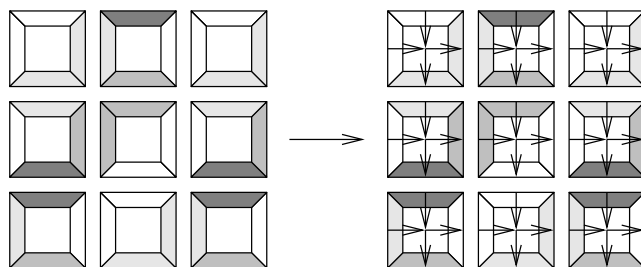


FIG. 3.4 – Illustration de la transformation “tuiles colorées” vers “tuiles fléchées”

Il est également simple de traduire des tuiles fléchées en formes. On transforme chaque tuile fléchée en une tuile de forme carrée, sauf que chaque côté contient un ergot, entrant ou sortant, dont la forme est spécifique et correspond à un type de flèche donné (Fig. 3.5).

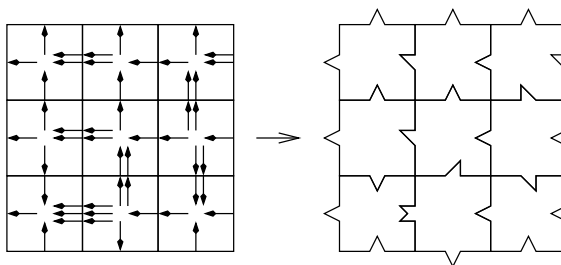


FIG. 3.5 – Illustration de la transformation “tuiles fléchées” vers “formes rationnelles”

Les tuiles obtenues ne se juxtaposent que si on a des ergots, l'un entrant et l'autre sortant, de même type, à l'instar des tuiles fléchées initiales.

La transformation qui traduit un ensemble de formes rationnelles en un ensemble de tuiles colorées est plus délicate. On commence par découper la forme de manière à obtenir des quadrilatères, qui sont des carrés, tous de même côté, intersectés avec un demi-plan. Ceci est possible puisque les sommets sont à coordonnées rationnelles. On colorie les bords de ces quadrilatères de telle sorte que l'on ne puisse les juxtaposer qu'en reconstruisant la pièce initiale; les bords extérieurs de la pièce initiale sont de couleur blanche. Enfin, on essaye de former des carrés complets avec ces pièces; on les fabrique en essayant toutes les combinaisons possibles qui aboutissent à un carré complet sans trou. Les nouveaux carrés peuvent donc avoir plusieurs couleurs sur un bord, on considère ce bord comme une nouvelle couleur. L'ensemble obtenu est le nouvel ensemble de tuiles colorées (Fig. 3.6 et Fig. 3.7).



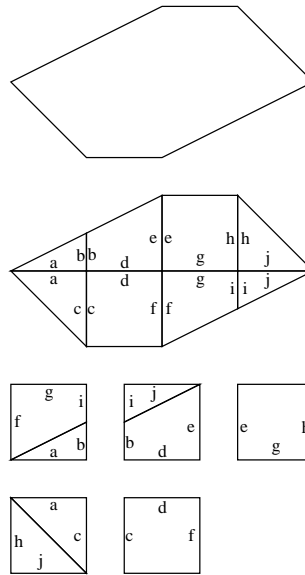


FIG. 3.6 – Illustration de la transformation “formes rationnelles” vers “tuiles colorées”

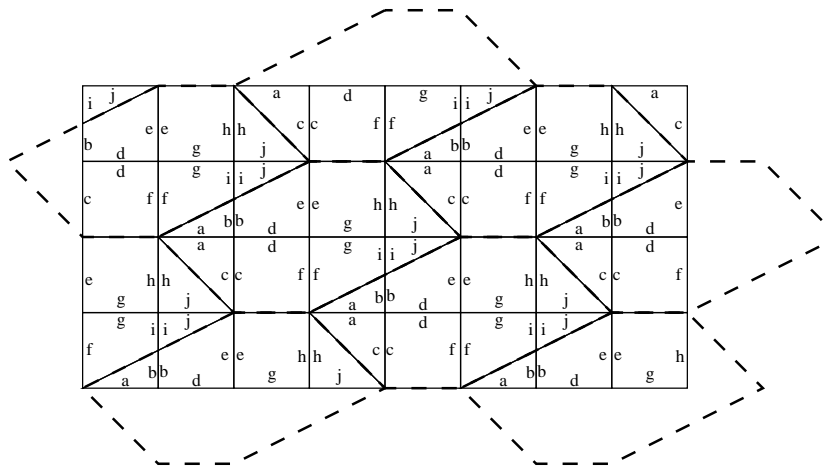


FIG. 3.7 – Pavage obtenu avec les tuiles de la figure 3.6

La transformation qui traduit un ensemble de tuiles colorées à une contrainte locale est simple. On prend un voisinage carré dont la taille est  $3(\lceil \log n \rceil)$ , où  $n$  est le nombre de tuiles colorées. Chaque tuile est codée par un motif carré de taille  $\lceil \log n \rceil$  dont la première ligne est le numéro de cette tuile. La contrainte locale n'est vérifiée que si les tuiles adjacentes ont les bonnes couleurs.

La transformation qui traduit une contrainte locale en un ensemble de tuiles colorées est un peu plus compliquée. On commence par la transformer, en agrandissant le voisinage pour qu'il soit carré de côté  $n > 1$ . Les couleurs seront les motifs rectangulaires de taille  $n \times (n+1)$  et  $(n+1) \times n$  qui vérifient la contrainte locale. On prendra comme ensemble de tuiles les quadruplets  $(h, b, g, d)$  tels que  $h$  et  $b$  aient  $n$  colonnes et  $n+1$  lignes, et que les  $n$  lignes inférieures de  $h$  soient égales aux  $n$  lignes supérieures de  $b$ , et que  $g$  et  $d$  aient  $n+1$  colonnes et  $n$  lignes, et que les  $n$  colonnes de gauche de  $d$  soient les  $n$  colonnes de droites de  $g$  (Fig. 3.8).  $\square$

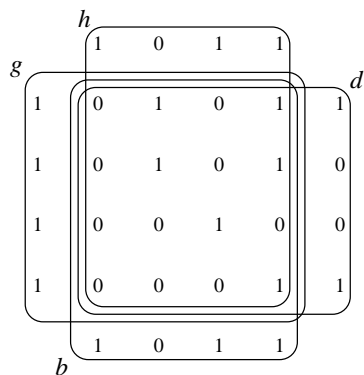


FIG. 3.8 – Illustration de la transformation “Contrainte locale” vers “tuiles colorées”

### 3.1.2.3 Problème de pavabilité

Un problème naturel sur les pavages est le problème de pavabilité du plan par un ensemble donné de tuiles.

#### Problème 1 (domino problem).

**Entrée:** Un ensemble de tuiles  $\tau$  ;

**Question:** Existe-t-il un pavage  $P$  du plan réalisé à l'aide des tuiles de  $\tau$  ?

**Théorème 8 (Berger [Ber66]).** *Le “domino problem” est indécidable.*

Comme signalé dans l'introduction de ce chapitre, ce problème est fondamental, puisqu'il permet de prouver l'indécidabilité des *Entscheidungsproblem*, et qu'aucune autre preuve n'existe à l'heure actuelle. De plus, la démonstration de ce théorème provient de la découverte d'un ensemble de tuiles dont aucun pavage du plan n'est périodique, ce que l'on croyait initialement impossible et donc on conjecturait que le *domino problem* était décidable. Ainsi, pour décider si un ensemble de tuiles admet un pavage du plan, on essayait, d'une part, de

former un motif périodique, et d'autre part, de construire des motifs de plus en plus grands. L'un de ses processus s'arrête obligatoirement, et l'on a la réponse au problème posé. La découverte de ce cas intermédiaire à donc complètement remis en question les idées de l'époque.

### 3.1.3 Périodicité et apériodicité

#### 3.1.3.1 Pavages périodiques

Une des manières les plus simples de paver le plan est de construire un pavage périodique. En effet, si, à l'aide de tuiles de Wang, on peut fabriquer un motif tel que la suite des couleurs est la même en haut et en bas, puis à droite et à gauche (Fig. 3.9), alors en les juxtaposant l'on peut paver tout le plan (Fig. 3.10).

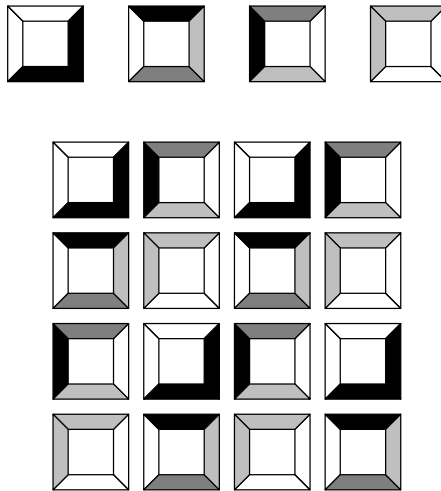


FIG. 3.9 – Un motif périodique...

Voici la définition formelle d'un pavage périodique :

**Définition 37 (pavage périodique).** *Un pavage  $P$  est périodique s'il existe deux entiers  $p$  et  $q$  tels que pour tout couple d'entiers  $(i, j)$ ,*

$$P(i, j) = P(i + p, j) = P(i, j + q)$$

Ainsi, après la résolution du *domino problem*, un autre problème se pose :

**Problème 2 (periodic domino problem).**

**Entrée:** *Un ensemble de tuiles  $\tau$  ;*

**Question:** *Existe-t-il un pavage  $P$  périodique du plan réalisé à l'aide des tuiles de  $\tau$  ?*

Gurevich et Koryakov ont montré dans [GK72] que ce problème est également indécidable. Ils ont aussi montré que l'ensemble des ensembles de tuiles qui pavent le plan périodiquement et l'ensemble des ensembles de tuiles qui ne pavent pas le plan sont récursivement inséparables, c'est-à-dire qu'il n'existe pas de fonction récursive, totale ou partielle, qui accepte toutes les valeurs du

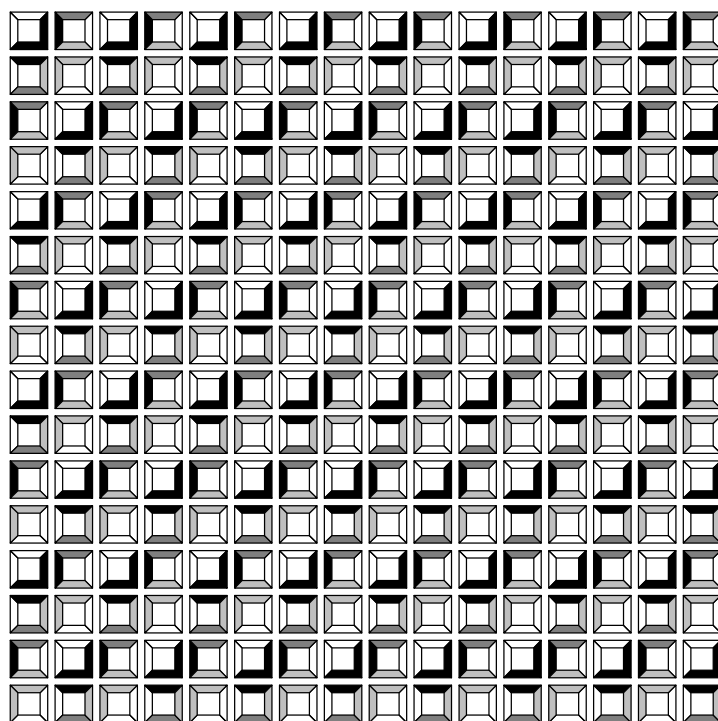


FIG. 3.10 – ... donne un pavage périodique du plan

premier et refuse toutes celles du second ; son comportement vis-à-vis des autres valeurs n'importe pas.

Cependant, de tels résultats n'ont été démontrés qu'avec la découverte de pavages non périodiques du plan.

### 3.1.3.2 Pavages quasipériodiques

Les premiers pavages aperiodiques construits furent des pavages fondés sur une construction autosimilaire. En effet, à l'aide des tuiles de base, on fabrique des pièces plus grosses qui ont la même forme (ou à peu près la même forme) que les tuiles de base du pavage. Ainsi, en répétant ce mécanisme, on peut construire des pièces de plus en plus grandes (Fig. 3.11). Ainsi, on peut obtenir des pièces

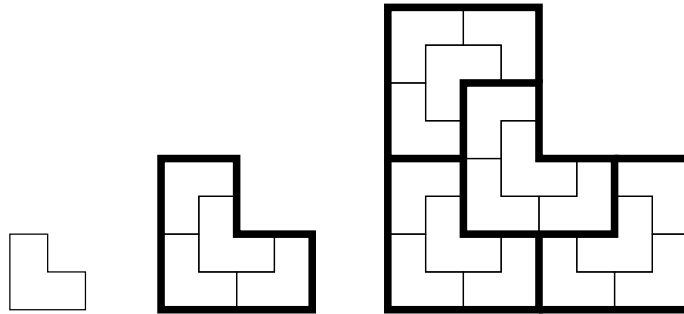


FIG. 3.11 – Un exemple de construction autosimilaire

de plus en plus grosses. Grâce à une extraction diagonale, on peut en déduire un pavage du plan (Fig. 3.12).

Un ensemble de tuiles introduit par Robinson dans [Rob71] permet de mieux comprendre et de simplifier les preuves d'indécidabilité des problèmes de pavabilité. C'est d'ailleurs sur cet ensemble que se fonde la construction de l'une de nos preuves.

Cet ensemble de tuiles est représenté figure 3.13, tel que l'avait présenté Robinson. La figure 3.14 est le même ensemble, mais représenté à l'aide de tuiles fléchées. Il permet de paver le plan à l'aide d'une construction autosimilaire. Les figures 3.15 et 3.16 montrent les niveaux deux et trois de la construction, le niveau un n'étant constitué que de la tuile  $\mathcal{D}$ . Passer d'un niveau au suivant consiste à placer la tuile  $\mathcal{D}$  au milieu, mettre à chaque coin un motif de la taille précédente, tourné comme montré figure 3.17.

Afin de mieux comprendre ce pavage, on ne représente souvent que les traits correspondant aux flèches doubles ( $\rightleftarrows$ ). Ainsi, seul le contour de chaque motif de la suite induite par la construction autosimilaire est représenté. On obtient alors la figure 3.18. Si on ne représente qu'un carré sur deux, on obtient la figure 3.19. On peut y voir que ce pavage contient des carrés de plus en plus grands, dont les côtés ne se croisent jamais. Ainsi, en faisant du calcul dans ces carrés de plus en plus grands (on a donc un espace de plus en plus grand pour calculer), on peut simuler une machine de Turing. C'est notamment sur cette idée que se fonde les preuves de l'indécidabilité des problèmes de pavabilité du plan.

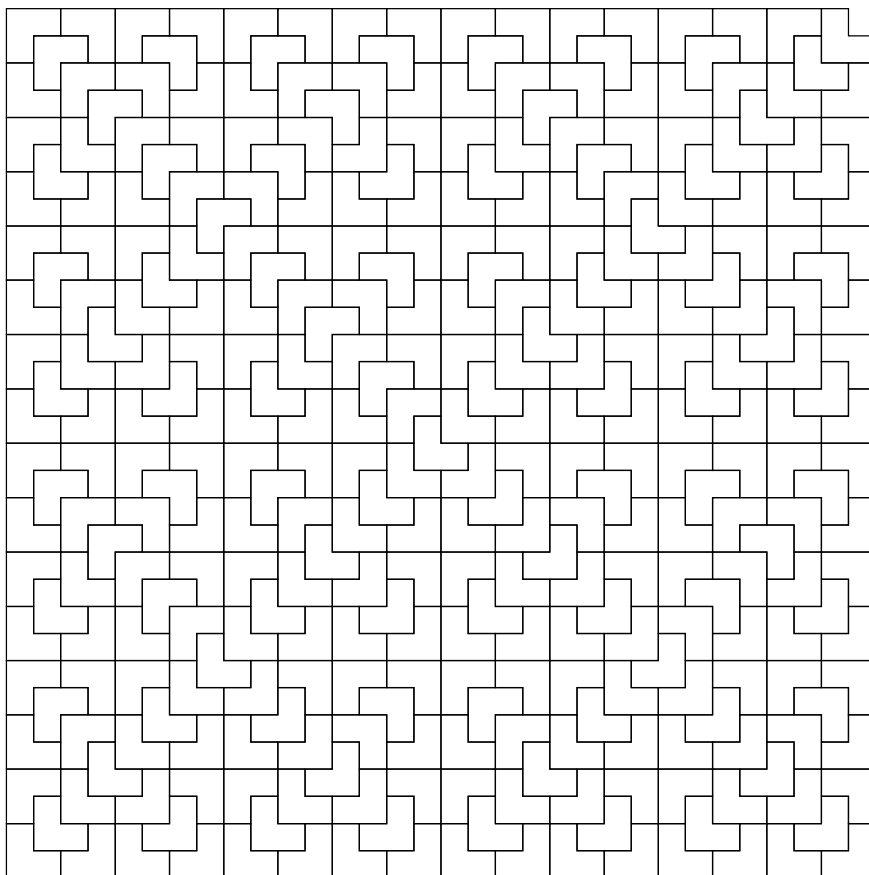


FIG. 3.12 – Pavage obtenu grâce à la construction de la figure 3.11

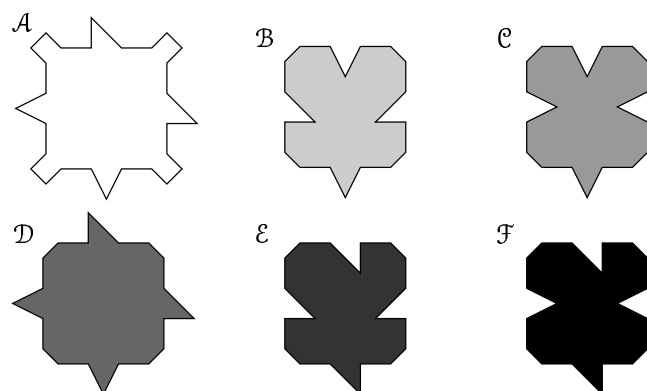


FIG. 3.13 – Ensemble de tuiles de Robinson (représentation originale) ; les niveaux de gris sont ajoutés pour faciliter la reconnaissance des pièces.

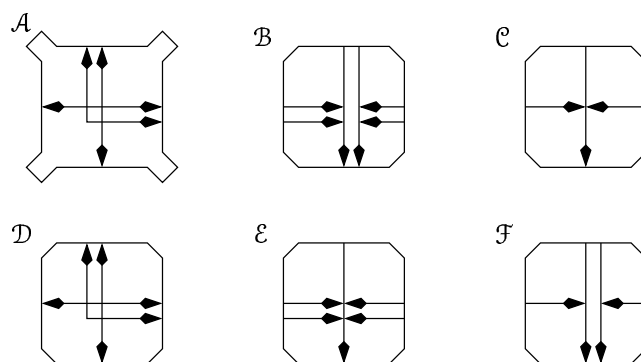


FIG. 3.14 – Ensemble de tuiles de Robinson (tuiles fléchées)

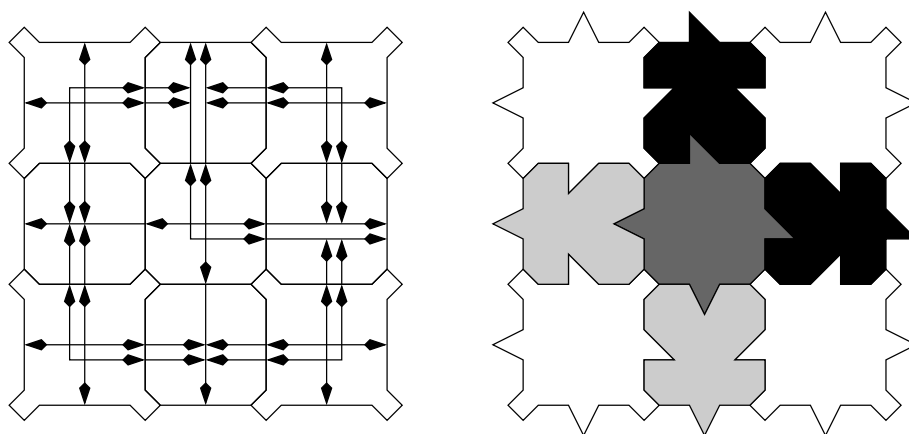


FIG. 3.15 – Deuxième niveau

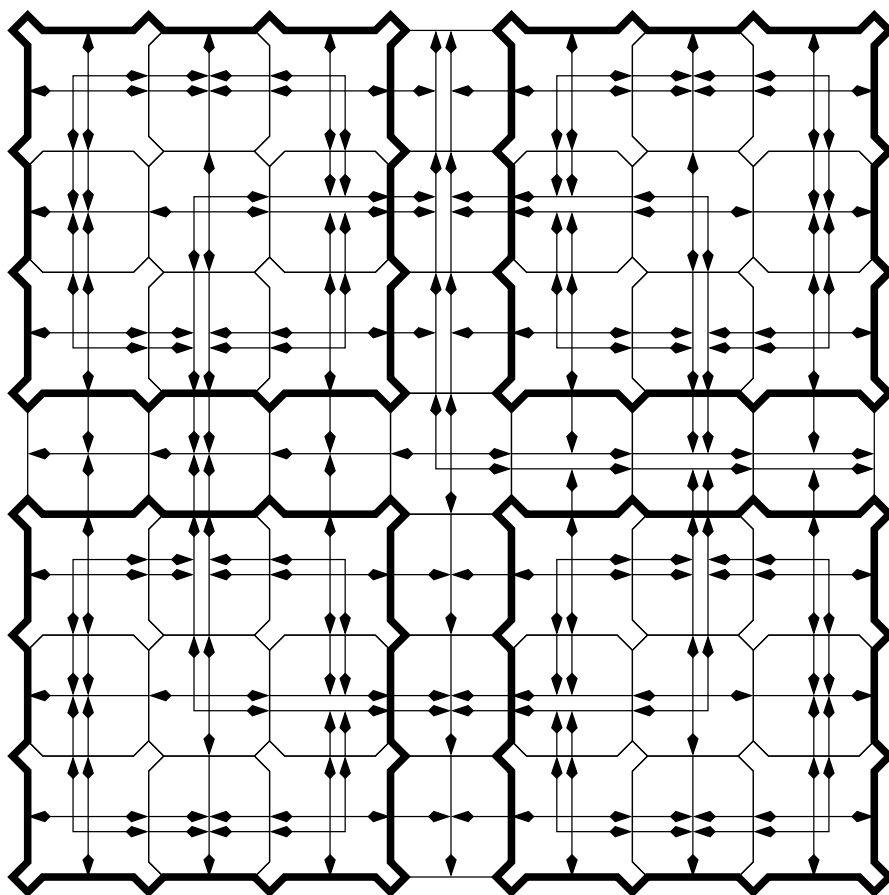


FIG. 3.16 – Troisième niveau



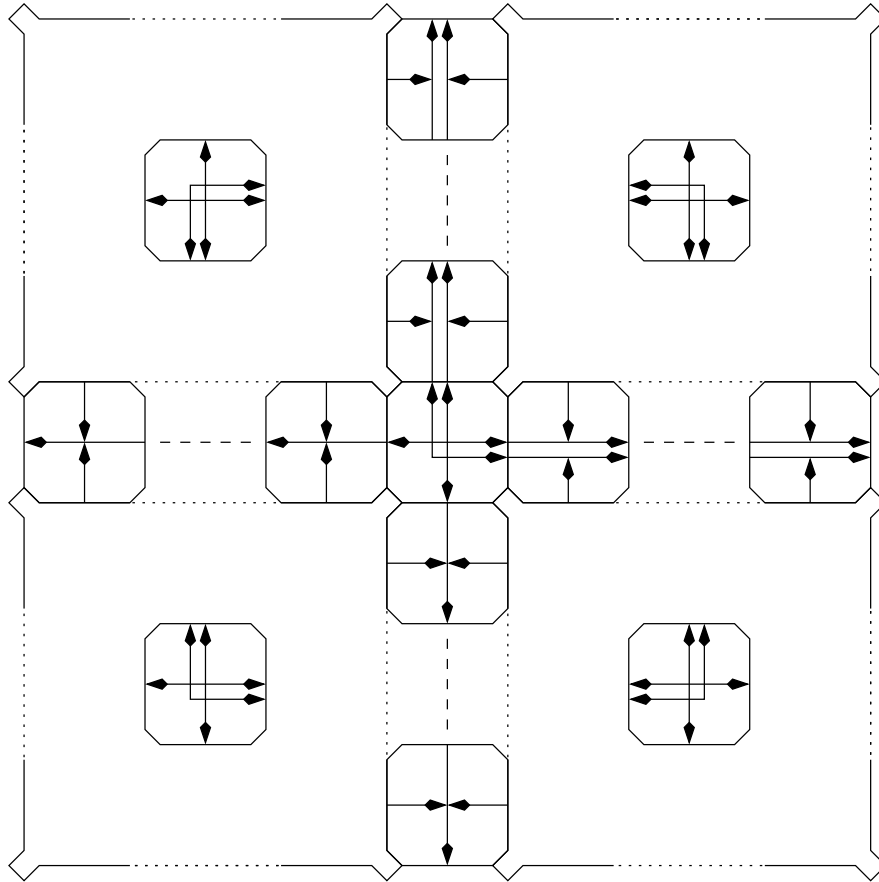


FIG. 3.17 – Passage d'un niveau à l'autre

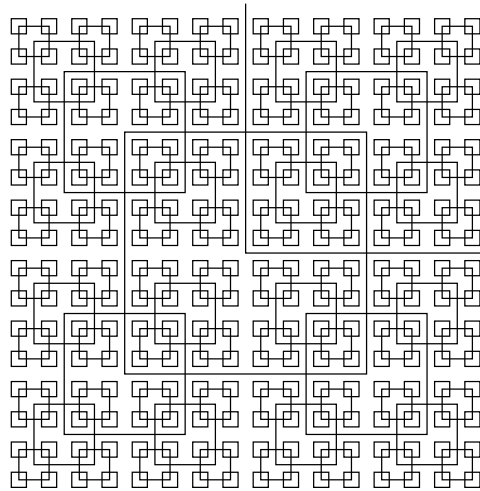


FIG. 3.18 – Vue simplifiée du pavage de Robinson

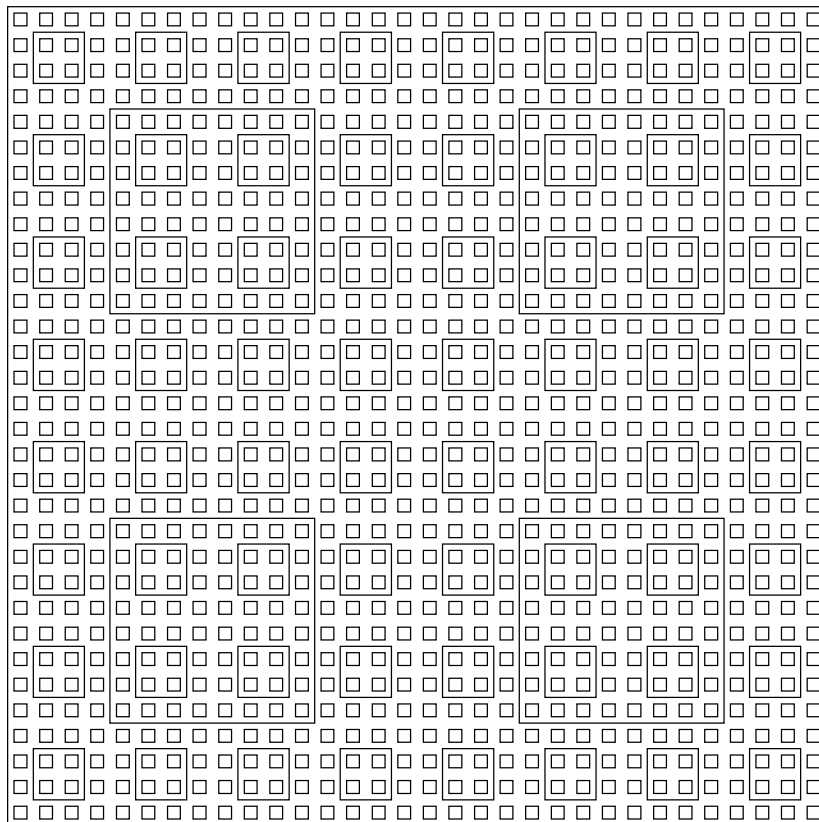


FIG. 3.19 – Vue dans laquelle on se représente qu'une taille de carré sur deux

## 3.2 Quantification de la quasipériodicité

Les pavages peuvent avoir une quasipériodicité plus ou moins complexe, c'est-à-dire qu'il faut une vue plus ou moins étendue du pavage pour ne pas le confondre avec un pavage apériodique. Par exemple, un pavage périodique a une quasipériodicité très simple ; celle d'un pavage fondé sur une construction autosimilaire l'est un peu moins. La *fonction de quasipériodicité* formalise cette notion de complexité.

Pour la définir, nous avons besoin de la notion de *motif* d'une configuration binaire.

**Définition 38 (motif).** *Un motif  $M$  est un couple  $(\mathcal{D}, f)$ , où  $\mathcal{D}$  est un sous-ensemble fini de  $\mathbb{Z}^2$  et  $f$  une fonction de  $\mathcal{D}$  dans  $\{0, 1\}$ . L'ensemble  $\mathcal{D}$  est appelé domaine du motif  $M$  et noté  $\text{Dom}(M)$ . Pour tout  $x \in \text{Dom}(M)$ , on note  $M(x) = f(x)$ . Dans le cas de motifs carrés, c'est-à-dire dont le domaine est  $\llbracket 1, a \rrbracket \times \llbracket 1, a \rrbracket$  pour  $a \in \mathbb{N}^*$ , on parlera de motif carré de taille  $a$ , et on notera  $a = |M|$ .*

*Si  $p$  est une configuration binaire, on dit que  $M$  est un sous-motif de  $p$  s'il existe un couple  $s \in \mathbb{Z}^2$  tel que :*

$$\forall x \in \text{Dom}(M), M(x) = p(s + x)$$

*et dans ce cas,  $s$  est appelée position du motif  $M$  dans  $p$ .*

**Définition 39 (fonction de quasipériodicité).**

*Soit  $P$  un pavage quasipériodique. La fonction de quasipériodicité  $f$  est définie de  $\mathbb{N}$  dans  $\mathbb{N}$ , et associe  $n$  à la taille minimale  $f(n)$  telle que tout motif carré de côté  $f(n)$  contient tous les motifs carrés de côté  $n$  présents dans le pavage (Fig. 3.21). On notera  $f = \mathcal{Q}_P$ .*

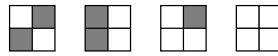


FIG. 3.20 – Un exemple : les seuls motifs carrés de côté 2 du pavage

Ainsi, un pavage périodique de période minimale  $p$  a comme fonction de quasipériodicité  $n \mapsto n + p - 1$ . Un pavage fondé sur une construction autosimilaire a une fonction de quasipériodicité linéaire ([Dur99]).

Dans la section suivante, nous allons voir quelles fonctions on peut atteindre avec des pavages.

### 3.2.1 Étude de la fonction de quasipériodicité

Dans le cadre de l'étude de la complexité structurelle des pavages, nous allons à présent essayer de cerner quelles fonctions on peut atteindre à l'aide de fonctions de quasipériodicité. Nous allons prouver que les fonctions croissantes "usuelles" peuvent être obtenues. Cela permet de montrer que la complexité permet de classer les pavages en une hiérarchie continue. D'autre part, on construit un pavage quasipériodique, mais dont la fonction de quasipériodicité est plus grande que toute fonction récursive. Ainsi, ce pavage est quasipériodique, mais on ne peut pas l'observer, ni le différencier d'un pavage non quasipériodique.

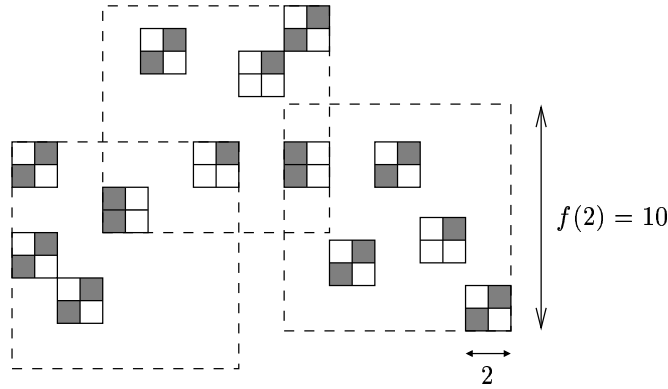


FIG. 3.21 – Un exemple : si les motifs carrés de côté 2 du pavage sont ceux de la figure 3.20, alors dans chaque fenêtre carrée de côté 10, tous ces motifs apparaissent, donc  $f(2)$  est au plus 10

**Définition 40 (constructible en temps).** Une fonction  $f$  est constructible en temps s'il existe une machine de Turing sur l'alphabet  $\{0, 1\}$  qui s'arrête après  $f(i)$  pas de calcul sur l'entrée  $1^i$ .

**Théorème 9.** Quelle que soit la fonction constructible en temps  $f$ , il existe un ensemble de tuiles  $\tau$  tel que, pour tout pavage  $p$  issu de  $\tau$ ,

$$\frac{(\S f(x) + 3)^2}{2} + 1 > Q_p(x) > \frac{(\S f(x) + 3)^2}{8} + 1$$

où

$$\S f(x) = \sum_{i=0}^x (i+1)f(i) .$$

De manière informelle, ce théorème stipule que pour toute fonction croissante constructible en temps, on peut trouver un ensemble de tuiles dont tous les pavages sont quasipériodiques, et dont la fonction de quasipériodicité est à peu près  $f$ .

*Preuve.* Soit  $M$  une machine de Turing vérifiant la définition de constructibilité en temps de  $f$ .

Pour cela, nous simulons l'évolution d'une machine de Turing en utilisant une construction semblable à celle décrite par Berger décrite dans [Ber66], celle qui fût utilisée pour démontrer l'indécidabilité du domino problem [Ber66, Rob71, AD96]. L'idée de la construction est d'utiliser le pavage de Robinson, et de simuler une machine de Turing dans chaque carré de la figure 3.19. Chaque carré n'utilise qu'une partie de son intérieur pour calculer, car il est gêné par les autres carrés. Cependant, la zone de calcul est de plus en plus grande, et donc, par extraction diagonale, on peut construire un diagramme espace-temps de la machine de Turing. Dans cette preuve, nous reprenons la même idée, mais nous représentons le diagramme espace-temps de la machine de Turing de manière différente afin de mieux contrôler les motifs qui y apparaissent.

Soient  $\{q_0, q_1, \dots, q_k\}$  l'ensemble des états de la machine  $M$ , et  $q_f$  son état final. Le diagramme espace-temps du calcul d'une machine de Turing sera la superposition des rubans de cette machine où on aura inséré une case contenant l'état de la machine juste avant la case où se trouve la tête. Par exemple, la configuration initiale sur l'entrée 111111 est  $q_011111$ .

Pour être sûr que de nouveaux petits motifs n'apparaissent pas plus tard dans le diagramme du calcul de  $M$ , ce qui augmenterait la valeur de la fonction de quasipériodicité, on entoure chaque caractère du ruban de la machine de Turing par plusieurs rangées de  $\#$ . A la  $n^{\text{ème}}$  étape, on remplace donc le diagramme par un autre,  $n$  fois plus gros, dans lequel chaque caractère ou état  $a$  est représenté par le motif carré de taille  $n$  représenté Fig. 3.22. On no-

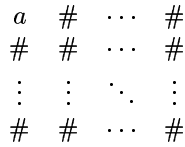


FIG. 3.22 – Diagramme  $\mathcal{S}_n(a)$  :  $n$  lignes et  $n$  colonnes

tera un tel motif  $\mathcal{S}_n(a)$ . Ainsi, la configuration précédente est représentée par  $\mathcal{S}_n(q_0)\mathcal{S}_n(1)\mathcal{S}_n(1)\mathcal{S}_n(1)\mathcal{S}_n(1)\mathcal{S}_n(1)\mathcal{S}_n(1)$ .

Dans chaque carré de calcul de la construction, on va superposer les diagrammes  $\mathcal{D}_n$  représentés figure 3.23, pour  $n$  variant de 0 à l'infini. Le diagramme  $n$  est constitué d'une ligne rappelant tous les motifs  $\mathcal{S}_n(a)$ , pour  $a$  état ou lettre, du diagramme espace-temps de la machine  $M$  sur l'entrée  $1^n$ , puis de la répétition de la première ligne. On ajoute une marge contenant  $1^n \diamond$  pour permettre de savoir à quelle étape on est lors du passage à l'étape suivante.

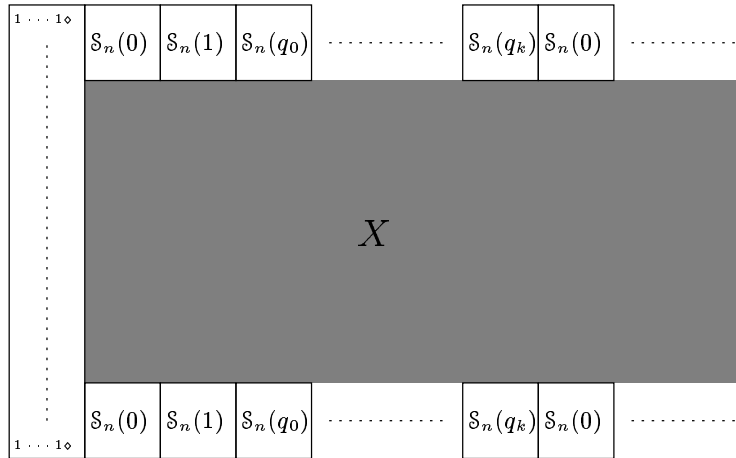


FIG. 3.23 – Diagramme  $\mathcal{D}_n$  – la zone grisée  $X$  contient le diagramme espace-temps de  $M$  avec pour entrée  $1^n$ , où chaque caractère ou état  $a$  est représenté par  $\mathcal{S}_n(a)$

Comme toutes les lettres et tous les états de la machine de Turing sont

énumérés à la première et dernière lignes du diagramme, on est sûr qu'aucun motif de taille  $n$ , autre que les motifs ne contenant que des  $\#$  et une lettre ou un état, n'apparaît dans ce diagramme. De même, comme cette liste est répétée au début et à la fin du diagramme, on est sûr qu'il faut une fenêtre contenant tout le calcul des diagrammes  $\mathcal{D}_1$  à  $\mathcal{D}_n$ , afin d'avoir exactement tous les motifs de taille  $n$ .

La construction utilisée pour passer du diagramme  $\mathcal{D}_n$  au diagramme  $\mathcal{D}_{n+1}$  est décrite figures 3.24 et 3.25. Le signal est initialisé par l'apparition d'un état final dans le diagramme. La figure 3.24 permet d'ajouter un 1 à la marge, et la figure 3.25 de créer la première bande du diagramme  $\mathcal{D}_{n+1}$ . Dans ces figures,

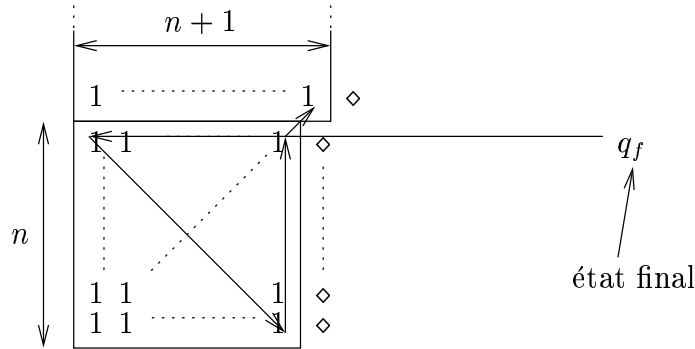


FIG. 3.24 – Construction permettant d'ajouter un 1 à la marge

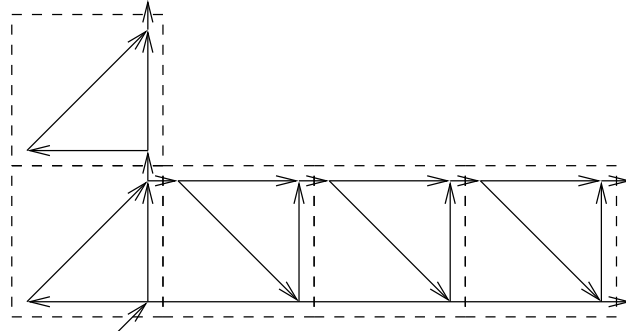


FIG. 3.25 – Construction permettant de créer la première bande du diagramme  $\mathcal{D}_{n+1}$

les flèches représentent un flux d'information, par exemple représenté par un certain type de flèches, lors de l'utilisation de tuiles fléchées.

Grâce à la construction de Berger, on sait que toute fenêtre de taille  $2^{2n+1} + 1$  contient les  $2^{n+2} - 3$  premières lignes de la superposition des diagrammes. Le fait que ce diagramme soit interrompu par les carrés intérieurs n'ajoute pas de nouveaux motifs puisque couper un  $\mathcal{S}_n(a)$  donne des  $\mathcal{S}_m(b)$ , pour  $m < n$ . Ainsi, pour obtenir tous les motifs de taille  $n$ , il faut les  $n$  premiers diagrammes  $\mathcal{D}_n$ . La taille de ces  $n$  premières étapes est  $\int f(x)$ . Ainsi, il faut et suffit une taille de

fenêtre égale à  $2^{2\ell+1} + 1$ , où  $\ell$  est le plus petit entier vérifiant  $2^{\ell+2} - 3 > \mathcal{J}f(n)$ . On en déduit que, si  $q$  est la fonction de quasipériodicité de ce pavage,

$$q(n) = 2^{2\ell+1} + 1$$

et donc que :

$$\frac{(\mathcal{J}f(n) + 3)^2}{2} + 1 > q(n) > \frac{(\mathcal{J}f(n) + 3)^2}{8} + 1$$

□

### 3.2.2 Pavages dont la quasipériodicité est non récursivement bornée

Nous avons prouvé que pour quasiment toute fonction récursive  $f$ , on peut construire un ensemble de tuiles tel que tous les pavages qu'il engendre sont quasipériodiques, et ont pour fonction de quasipériodicité à peu près  $f$ . A présent, nous construisons un ensemble de tuiles qui ne produit que des pavages quasipériodiques, et dont la fonction de quasipériodicité est plus grande que toute fonction récursive.

**Théorème 10.** *Il existe un ensemble de tuiles  $\tau_0$  tel que pour tout pavage  $p$  produit par  $\tau_0$ ,  $p$  est quasipériodique, et il n'existe pas de fonction récursive qui soit une borne supérieure de  $\mathcal{Q}_p$ .*

Ce théorème exprime que la régularité de tout pavage produit par  $\tau_0$  ne peut pas être observée.

*Preuve.* Soit  $\mathbb{K}$  un sous-ensemble non récursif, mais récursivement énumérable de  $\{0, 1\}^*$  (par exemple, l'ensemble des paires  $(x, y)$  telles que la machine de Turing dont le codage en binaire est  $x$  s'arrête sur l'entrée dont le codage en binaire est  $y$ ).

L'idée de la preuve est de fabriquer un ensemble de tuiles qui émule une machine de Turing qui énumère  $\mathbb{K}$ . Si l'on peut calculer la taille  $t$  d'une fenêtre dans laquelle tous les motifs de taille  $n$  apparaissent, il suffit alors de faire  $t$  pas de calcul pour voir apparaître tous les motifs de taille  $n$ , et donc savoir si un mot donné de taille  $n$  a été énuméré par la machine. On pourrait, dans ce cas, décider  $\mathbb{K}$ .

Soit  $M$  une machine de Turing sur l'alphabet  $\{0, 1, \#\}$ , à deux rubans, le premier étant un ruban de travail, et qui, sur le second, énumère les éléments de  $K$  séparés par des  $\#$  et en maintenant les éléments classés par ordre croissant.

Par exemple, si l'énumération de  $K$  est  $23, 67, 45, 12, 36, 52, 38, \dots$ , la machine écrit :

```

23
23#67
23#45#67
12#23#45#67
12#23#36#45#67
12#23#36#45#52#67
12#23#36#38#45#52#67
⋮

```

Soit  $\tau_0$  l'ensemble de tuiles permettant, grâce à la construction de Berger ([AD96]), d'émuler la machine de Turing  $M$ . Soit  $p$  un pavage produit par  $\tau_0$ . Supposons, par l'absurde, qu'il existe une fonction récursive  $f$  telle que pour tout entier  $x$ ,  $f(x) > Q_p(x)$ . Montrons que dans ce cas, on peut décider  $K$ .

Grâce à la construction de Robinson, on sait qu'il existe une fonction récursive  $\rho$  telle que  $n$  cellules sur le second ruban sont représentées dans un motif carré de taille  $\rho(n)$  issu de  $p$ .

Montrons que, sous l'hypothèse de l'existence de  $f$ ,  $K$  est récursif. Soit  $x \in K$  un mot de taille  $\ell$ , qui est donc énuméré par  $M$ . Le mot  $x$  est écrit à au plus  $b = 1 + \sum_{i=0}^x 1 + \log i$  cases du début du second ruban, puisque  $M$  écrit les entiers par ordre croissant. Ainsi, il y a, dans le pavage  $p$ , un motif carré  $C$  de taille  $a = \rho(b)$  qui contient la représentation des  $b$  premières cases du second ruban. Comme tous les motifs carrés de taille  $a$  se trouvent dans tout motif de taille  $f(a)$ , un mot  $y$  de taille inférieure à  $\ell$  est énuméré par  $M$  si et seulement si  $\#y\#$  apparaît dans  $C$ , et donc dans toute fenêtre carrée de taille  $f(a)$ . Comme grâce à la construction de Robinson, on sait construire un motif carré de taille  $f(a)$  qui peut s'étendre en un pavage du plan, et que tout pavage  $p'$  créé à l'aide des tuiles de  $\tau_0$  contient les mêmes motifs que  $p$  (voir [Dur99]), alors, on peut ainsi décider si  $y$  est dans  $K$ , ce qui est absurde.  $\square$

### 3.3 Variations sur le théorème de Rice

Dans cette section, nous allons tenter de démontrer une variante adaptée aux pavages du théorème démontré par Rice sur les programmes, comme quoi l'ensemble des machines de Turing calculant une fonction ayant une propriété particulière est indécidable. On espère ainsi aborder la complexité algorithmique des pavages. En effet, les notions de calculabilité sont très présentes dans plusieurs résultats concernant les pavages. Ainsi, outre l'indécidabilité de différents problèmes, il existe un ensemble de tuile dont tout pavage produit n'est pas récursif (il n'existe pas de programme qui, recevant en entrée les coordonnées d'une case, donne quelle tuile il faut y mettre). De plus, dans la section précédente, on a donné un ensemble de tuiles dont la fonction de quasipériodicité n'est pas récursivement bornée.

#### 3.3.1 Théorème de Rice pour les programmes

Soit  $(M_i)_{i \in \mathbb{N}}$  une énumération calculable des machines de Turing, et  $\varphi_i$  la fonction calculée par la machine  $M_i$ . Par souci de complétude, on redémontre ici le théorème de Rice :

**Théorème 11 (Théorème de Rice).** *Soit  $\mathcal{C} \subset \mathbb{N}^{\mathbb{N}}$  une classe de fonctions. Alors l'ensemble*

$$M_{\mathcal{C}} = \{i \in \mathbb{N}, \varphi_i \in \mathcal{C}\}$$

*est vide, plein ou indécidable (many-one complet).*

*Preuve.* On notera  $q(x) = \perp$  pour signifier que la fonction  $q$  ne s'arrête pas sur l'entrée  $x$ .

Supposons que  $M_{\mathcal{C}}$  ne soit ni vide ni plein. Soit  $T$  la fonction qui ne s'arrête nulle part. Si  $T \in M_{\mathcal{C}}$ , soit  $\psi \notin M_{\mathcal{C}}$ , et sinon, soit  $\psi \in M_{\mathcal{C}}$ . La fonction  $\psi$  existe car  $M_{\mathcal{C}}$  n'est ni vide ni plein.



Supposons par l'absurde que  $M_{\mathcal{C}}$  est décidable. Soit  $f$ , la fonction suivante :

$$f(x, y) = \begin{cases} \perp & \text{si } \varphi_x(x) = \perp \\ \psi(y) & \text{sinon} \end{cases} .$$

Elle est calculable puisqu'elle commence à calculer  $\varphi_x(x)$ , puis  $\psi(x)$  si ce premier calcul s'arrête. En appliquant le théorème  $s_n^m$ , on en déduit qu'il existe une fonction calculable  $g$  telle que

$$\varphi_{g(x)} = \begin{cases} T & \text{si } \varphi_x(x) = \perp \\ \psi & \text{sinon} \end{cases} .$$

Ainsi, si  $T \in M_{\mathcal{C}}$ , on a  $g(x) \in M_{\mathcal{C}} \iff \varphi_x(x) = \perp$ , et si  $T \notin M_{\mathcal{C}}$ ,  $g(x) \notin M_{\mathcal{C}} \iff \varphi_x(x) = \perp$ . Par conséquent,  $M_{\mathcal{C}}$  est many-one réduit à  $\{x, \varphi_x(x) = \perp\}$  ou son complémentaire qui sont tous deux indécidables. L'ensemble  $M_{\mathcal{C}}$  est donc many-one complet.  $\square$

### 3.3.2 Théorème de Rice pour les pavages

Afin de ne pas buter sur des problèmes de comparaison de pavages du plan formé de tuiles provenant d'ensembles de tuiles différentes, nous utilisons la notion de pavabilité par contrainte locale. Si  $c$  est une contrainte locale, on notera  $\mathcal{P}(c)$  l'ensemble des configurations binaires produites par  $c$ .

Soit  $c$  une contrainte locale. On définit le problème suivant :

**Problème 3** (PCLI( $c$ )).

**Entrée:** Une contrainte locale  $\nu$  ;

**Question:** La contrainte  $\nu$  produit-elle exactement les mêmes configurations binaires que  $c$ , c'est-à-dire  $\mathcal{P}(\nu) = \mathcal{P}(c)$  ?

On appellera *triviale* toute contrainte locale dont la fonction de contrainte vaut 0 partout.

**Théorème 12 (Théorème de Rice pour les pavages).**

Si  $c$  est une contrainte locale non triviale, le problème PCLI( $c$ ) est indécidable, plus précisément  $\Sigma_1$ .

*Preuve.* Sans perte de généralité, on peut ne considérer que les contraintes locales dont le voisinage est de la forme  $\llbracket 1, a \rrbracket \times \llbracket 1, a \rrbracket$ , où  $a$  est un entier strictement positif, puisqu'il suffit de prendre le plus grand carré contenant le voisinage, puis de le translater de telle sorte que tous ses points aient des coordonnées strictement positives.

Montrons que ce problème est récursivement énumérable. Pour cela, on montre que l'ensemble des paires de contraintes locales qui produisent les mêmes configurations binaires est récursivement énumérable.

Soit  $f$  un programme qui, étant donnée une paire de contraintes locales  $(\mu, \nu)$ , effectue les étapes suivantes. Si  $\llbracket 1, m \rrbracket \times \llbracket 1, m \rrbracket$  est le domaine de  $\mu$ , et  $\llbracket 1, n \rrbracket \times \llbracket 1, n \rrbracket$ , celui de  $\nu$  :

- si  $\mu$  est triviale alors  $f$  s'arrête ;
- pour tout motif interdit  $F$  de domaine  $\llbracket 1, m \rrbracket \times \llbracket 1, m \rrbracket$  de  $\mu$  faire :
  - $i \leftarrow 0$
  - répéter (boucle  $\alpha$ )

- vérifier si tous les motifs de domaine  $\llbracket 1, m + 2i \rrbracket \times \llbracket 1, m + 2i \rrbracket$  dont le sous-motif central est  $F$  sont interdits pour  $\nu$  ;
- si oui, sortir de la boucle et incrémenter  $i$  ;
- fin répéter
- fin pour

Supposons que  $\mu$  n'est pas triviale, et supposons que  $f(\mu, \nu)$  s'arrête. Soit  $P$ , une configuration binaire qui ne vérifie pas  $\mu$ . Alors, elle contient un motif interdit  $F$  de domaine  $\llbracket 1, m \rrbracket \times \llbracket 1, m \rrbracket$ , à la position  $(x, y)$ . Le programme sort de la boucle  $\alpha$  alors qu'il traite  $F$  ; cela montre qu'il existe un entier  $i$  tel que tout motif carré de taille  $m + 2i$ , dont le sous-motif central est  $F$ , est interdit pour  $\nu$ . Ainsi, le motif  $P$  de position  $(x - i, y - i)$  est interdit pour  $\nu$ . Ceci implique que  $P$  ne vérifie pas  $\nu$ . On vient de montrer que toute configuration binaire non produite par  $\mu$  ne vérifie pas  $\nu$ .

Réciproquement, supposons que  $f(\mu, \nu)$  ne s'arrête pas. Alors, il existe un motif  $F$  pour lequel la boucle  $\alpha$  ne s'arrête jamais. On en déduit que pour tout entier  $i$ , il existe un motif de domaine  $\llbracket 1, m + 2i \rrbracket \times \llbracket 1, m + 2i \rrbracket$  dont le sous-motif central est  $F$  vérifiant  $\nu$ . Ainsi, par extraction diagonale, il existe une configuration  $R$  qui contient  $F$  et qui est produite par  $\nu$ . La configuration binaire  $R$  vérifie  $\nu$  mais pas  $\mu$ .

Ainsi, on vient de prouver que  $f(\mu, \nu)$  s'arrête si et seulement si toutes les configurations binaires qui vérifient  $\nu$  vérifient aussi  $\mu$ . Ceci est aussi vrai quand  $\mu$  est triviale. Soit  $M$  une machine de Turing qui, sur l'entrée  $(\mu, \nu)$ , calcule  $f(\mu, \nu)$ , puis  $f(\nu, \mu)$ . Cette machine s'arrête si et seulement si  $\mu$  et  $\nu$  produisent les mêmes configurations binaires. L'ensemble des paires de contraintes locales qui produisent les mêmes configurations binaires est récursivement énumérable.

Soit donc  $c$  une contrainte locale non triviale, dont le voisinage est  $\llbracket 1, a \rrbracket \times \llbracket 1, a \rrbracket$  et la fonction de contrainte  $\delta$ . Il existe donc un motif carré  $F$  tel que  $\delta(F) = 1$ , c'est-à-dire que :

$$\delta(F(1, 1), f(1, 2), \dots, F(|F|, |F|)) = 1$$

Toute configuration binaire contenant le motif  $F$  n'est donc pas produite par  $c$ .

Montrons que l'ensemble des contraintes locales qui produisent les mêmes pavages que  $c$  est complet pour la réduction many-one. Pour cela, nous allons réduire le domino problem à cet ensemble.

Soit  $\nu$ , une contrainte locale de domaine  $\llbracket 1, l \rrbracket \times \llbracket 1, l \rrbracket$ .

Pour tout motif carré  $M$ , nous noterons  $\widehat{M}$  le motif carré de taille  $|F|(2|M| + 1)$  illustré par la figure 3.26, et construit de la manière suivante. Le motif  $\widehat{M}$  est en fait une juxtaposition de  $2|M| + 1$  par  $2|M| + 1$  motifs carrés de taille  $|F|$ . Les motifs sont

- pour tout  $i \in \llbracket 0, |M| \rrbracket$ , et tout  $j \in \llbracket 0, |M| \rrbracket$ , le motif carré de taille  $|F|$ , situé à la position  $((2i + 1)|F| + 1, (2j + 1)|F| + 1)$  vaut  $F$  ;
- pour tout  $i \in \llbracket 1, |M| \rrbracket$ , et tout  $j \in \llbracket 1, |M| \rrbracket$ , le motif carré de taille  $|F|$  situé à la position  $(2i|F| + 1, 2j|F| + 1)$  est rempli de  $M(i, j)$ .

Pour toute configuration binaire  $p$ , définissons la configuration binaire  $\hat{p}$  de la manière suivante : pour tout  $(m, n) \in \mathbb{Z}^2$ , et tout  $(i, j) \in \llbracket 1, |F| \rrbracket \times \llbracket 1, |F| \rrbracket$  :

$$\hat{p}(2n|F| + i, 2m|F| + j) = p(n, m)$$

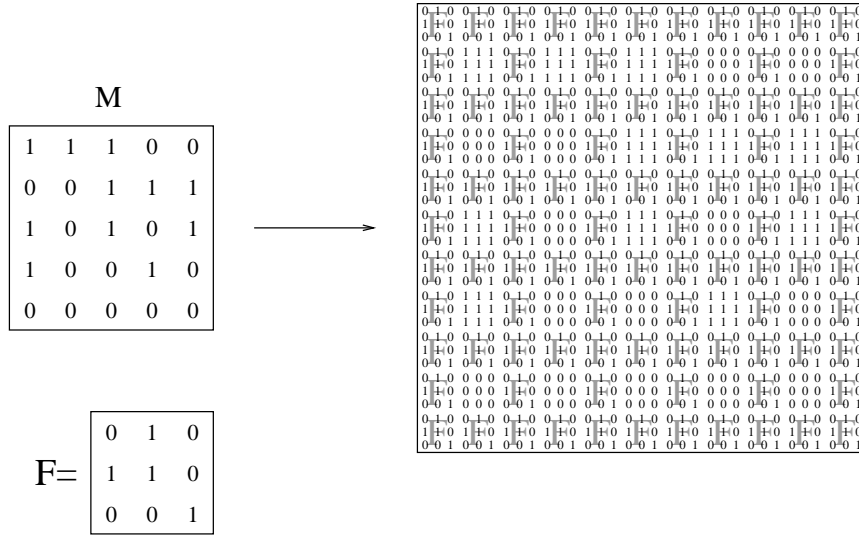


FIG. 3.26 – Un exemple de motif  $M$ , et le motif  $\hat{M}$  associé

et

$$\hat{p}(n|F| + i, (2m + 1)|F| + j) = \hat{p}((2n + 1)|F| + i, m|F| + j) = F(i, j).$$

Cette définition est une extension aux configurations binaires de la définition correspondante pour les motifs. Enfin, définissons la contrainte locale  $\hat{\nu}$  de domaine  $\llbracket 1, |F|(2l + 1) \rrbracket \times \llbracket 1, |F|(2l + 1) \rrbracket$  et de fonction de contrainte  $f$  qui ne vaut 0 que sur les motifs  $M$  définis ainsi :

- i)  $M$  est un motif produit par  $c$  (c'est-à-dire dont tous les sous-motifs annulent la fonction de contrainte);
- ii) si  $N$  est motif carré de taille  $l + 1$  produit par  $\nu$ ,  $M$  est un sous-motif de  $\hat{N}$ .

Toute configuration binaire produite par  $c$  vérifie  $\hat{\nu}$ , grâce à la règle *i*).

De plus, si  $\nu$  produit une configuration binaire  $t$ , d'après la règle *ii*),  $\hat{\nu}$  produit la configuration binaire  $\hat{t}$ .

Réciproquement, si  $\hat{\nu}$  produit une configuration binaire  $q$  qui ne vérifie pas  $c$ , cela signifie qu'elle contient un motif  $N$  qui viole la règle *i*), et donc qui vérifie la règle *ii*). Ce motif contient donc le motif  $F$  plusieurs fois, et donc les motifs  $N'$  de  $d$ , dont la position est celle de  $N$  décalée de 1 dans l'une des quatre directions, contiennent aussi le motif  $F$  et, par conséquent, violent la règle *i*). De proche en proche, on en déduit que tous les motifs de  $q$  vérifient la règle *ii*), et par conséquent qu'il existe un pavage  $t$  produit par  $\nu$  tel que  $q = \hat{t}$ .

On en déduit que  $\nu$  produit une configuration binaire si et seulement si  $\hat{\nu}$  ne produit pas les mêmes configurations binaires que  $c$ . Comme on peut construire  $\hat{\nu}$  récursivement en fonction de  $\nu$ , on en déduit que l'ensemble des contraintes locales qui produisent les mêmes configurations binaires que  $c$  est many-one complet, et donc non récursif. □

La même preuve sert à prouver le résultat suivant : soit  $c$  une contrainte locale. On définit le problème suivant :

**Problème 4** ( $\text{PCLN}(c)$ ).

**Entrée:** Une contrainte locale  $\nu$  ;

**Question:** Les configurations binaires produites par  $\nu$  sont-elles aussi produites par  $c$ , c'est-à-dire  $\mathcal{P}(\nu) \subset \mathcal{P}(c)$  ?

**Théorème 13 (Variante du théorème de Rice pour les pavages).**

Si  $c$  est une contrainte locale non triviale, le problème  $\text{PCLN}(c)$  est indécidable, mais récursivement énumérable.

*Preuve.* En ce qui concerne l'énumérabilité, on a construit dans la preuve précédente un programme  $f$  telle que pour toutes contraintes locales  $\mu$  et  $\nu$ ,  $f(\mu, \nu)$  s'arrête si et seulement si toutes les configurations binaires qui vérifient  $\nu$ , vérifient aussi  $\mu$ , soit  $\mathcal{P}(\nu) \subset \mathcal{P}(\mu)$ . Cette fonction permet donc bien d'énumérer récursivement les solutions du problème  $\text{PCLN}(c)$ .

Toujours dans la preuve du théorème précédent, à partir de la contrainte locale  $\nu$ , on construit la contrainte locale  $\hat{\nu}$  telle que : si  $\nu$  produit une configuration binaire alors  $\hat{\nu}$  produit strictement plus de configurations que  $c$ , soit  $\mathcal{P}(c) \subsetneq \mathcal{P}(\hat{\nu}) \implies \hat{\nu} \notin \text{PCLN}(c)$ ; si  $\nu$  n'en produit pas, alors  $\hat{\nu}$  produit les mêmes configurations que  $c$ , soit  $\mathcal{P}(c) = \mathcal{P}(\hat{\nu}) \implies \hat{\nu} \in \text{PCLN}(c)$ . On a donc bien réduit le *domino problem* à  $\text{PCLN}(c)$ .  $\square$

Ces résultats ne sont que des résultats partiels. On aimerait prouver un résultat plus fort :

**Problème Ouvert 4.** Existe-t-il une bonne définition de "trivial" telle que pour tout ensemble  $\mathcal{E}$ , non "trivial", d'ensembles de configurations binaires, savoir si l'ensemble des configurations binaires produites par une contrainte locale  $\tau$  appartient à  $\mathcal{E}$  est indécidable ?

Cette définition de "trivialité" n'est a priori pas aussi simple que celle du théorème précédent. Par exemple, savoir si une contrainte locale produit au moins la configuration binaire ne contenant que des 1 est décidable. Ainsi, pour tenter de prouver un résultat similaire au théorème précédent, mais en demandant si  $\mathcal{P}(c) \subset \mathcal{P}(\nu)$  au lieu de l'inclusion inverse, on est donc confronté au même problème de définition de "trivialité".

### 3.3.3 Théorème de Rice pour les automates cellulaires

L'analogie de définition entre contraintes locales et automates cellulaires permet de traduire la preuve précédente afin de l'appliquer aux automates cellulaires.

Si  $f$  est un automate cellulaire de rayon  $r$  et  $M$  est un motif carré de taille  $s > 2r$ , alors  $f(M)$  est le motif carré de taille  $s - 2r$  suivant :

$$\forall i, j \in \llbracket 1, s - 2r \rrbracket, f(M)(i, j) = \lambda(M(i, j), \dots, M(i + 2r + 1, j + r2 + 1))$$

où  $\lambda$  est la règle locale de  $f$ .

Considérons le problème suivant : étant donné un automate cellulaire  $f$  de dimension 2 admettant  $\{0, 1\}$  comme ensemble d'états

**Problème 5** (ELACI( $f$ )).

**Entrée:** un automate cellulaire  $g$  de dimension 2 sur  $\{0, 1\}$  ;

**Question:** l'automate  $g$  a-t-il le même ensemble limite que  $f$ , c'est-à-dire  $\Omega(f) = \Omega(g)$  ?

**Théorème 14.** Soit  $f$  un automate cellulaire à deux états (0 et 1), dont l'ensemble limite n'est pas l'ensemble de toutes les configurations. Le problème ELACI( $f$ ) est indécidable.

*Preuve.* Comme l'automate est à deux états, les configurations du plan sont en fait des configurations binaires. On reprendra aussi la notion de motif binaire.

Un ensemble limite d'un automate cellulaire est un sous-shift (ensemble fermé pour la topologie de Cantor, invariant par translation). Comme l'ensemble limite de l'automate cellulaire  $f$  n'est pas plein, il admet un motif interdit  $G$  de taille  $2a + 1$ , c'est-à-dire un motif qui n'apparaît dans aucune configuration de  $\Omega(f)$ . Il existe un motif  $F$  de taille  $2b + 1$ , qui contient  $G$  (qui est donc aussi interdit) tel que  $f(F)$  n'est pas le sous-motif central de  $F$ . En effet, sinon, on aurait des motifs de plus en plus grands contenant  $G$  et "invariant" par  $f$ , et on en déduirait, par extraction diagonale, un pavage invariant par  $f$ , et donc dans l'ensemble limite, mais contenant  $G$ , ce qui est absurde.

Quitte à prendre un motif plus grand, on suppose le motif  $F$  carré.

Montrons que l'ensemble des automates cellulaires dont l'ensemble limite est le même que celui de  $f$  est complet pour la réduction many-one. Pour cela, nous allons de nouveau réduire le domino problem à cet ensemble.

Soit  $\nu$ , une contrainte locale de domaine  $\llbracket 1, l \rrbracket \times \llbracket 1, l \rrbracket$ . Soit  $r$ , le rayon de l'automate cellulaire  $f$ . Pour tout motif carré  $M$ , et toute configuration binaire  $p$ , reprenons les notations  $\hat{M}$  et  $\hat{p}$  de la preuve précédente : le motif  $\hat{M}$  est une juxtaposition de  $2|M| + 1$  par  $2|M| + 1$  motifs de taille  $|F|$  (Fig. 3.26), les motifs étant :

- pour tout  $i \in \llbracket 0, |M| \rrbracket$ , et tout  $j \in \llbracket 0, |M| \rrbracket$ , le motif carré de taille  $|F|$ , situé à la position  $((2i + 1)|F| + 1, (2j + 1)|F| + 1)$  vaut  $F$  ;
- pour tout  $i \in \llbracket 1, |M| \rrbracket$ , et tout  $j \in \llbracket 1, |M| \rrbracket$ , le motif carré de taille  $|F|$  situé à la position  $(2i|F| + 1, 2j|F| + 1)$  est rempli de  $M(i, j)$  ;

la configuration binaire  $\hat{p}$  est définie de la manière suivante : pour tout  $(m, n) \in \mathbb{Z}^2$ , et tout  $(i, j) \in \llbracket 1, a \rrbracket \times \llbracket 1, a \rrbracket$ ,

$$\hat{p}(2n|F| + i, 2m|F| + j) = p(n, m)$$

et

$$\hat{p}(n|F| + i, (2m + 1)|F| + j) = \hat{p}((2n + 1)|F| + i, m|F| + j) = F(i, j).$$

Enfin, définissons l'automate cellulaire  $\hat{\nu}$  à deux états dont le rayon  $r'$  est  $\max(r, |F|(l + 1))$  et de règle locale  $\lambda$  qui, quand elle reçoit en entrée un motif  $M$ ,

- i) conserve la lettre  $(0, 0)$  s'il existe un motif carré  $N$  de taille  $l + 1$  produit par  $\nu$  tel que  $M$  est sous-motif de  $\hat{N}$  ;
- ii) applique la fonction locale de  $f$  sinon (ce qui est possible car  $r' \geq r$ ).

Toute configuration binaire  $d$  de  $\Omega(f)$  ne contient pas le motif  $F$ , et par conséquent, l'automate cellulaire  $\hat{\nu}$  agit comme l'automate  $f$  grâce au point ii).

De plus, pour tout  $n$ ,  $f^n(d) \in \Omega(f)$  et ne contient pas  $F$  non plus. Ainsi, pour tout  $i$ ,  $f^i(d) = \hat{\nu}^i(d)$ , et donc  $d \in \Omega(\hat{\nu})$ . On en déduit que :

$$\Omega(f) \subset \Omega(\hat{\nu}) . \quad (3.1)$$

Pour toute configuration binaire  $c$  produite par  $\nu$ , alors  $\hat{c}$  est dans l'ensemble limite de  $\hat{\nu}$ , puisque  $\lambda$  conserve les lettres centrales des sous-motifs des  $\hat{N}$ , pour  $N$  produit par  $\nu$ , et que dans ce cas, tous les sous-motifs de  $\hat{c}$  vérifient cette hypothèse. On en déduit que l'inclusion de l'équation (3.1) est stricte si  $\nu$  produit au moins une configuration binaire.

Réciproquement, si  $\nu$  ne produit aucune configuration binaire, montrons que cette inclusion est en fait une égalité. Soit  $c$  une configuration binaire de  $\Omega(\hat{\nu})$ . Il existe donc une suite infinie de configurations  $(x_i)_{i \in \mathbb{N}}$  de  $\Omega(\hat{\nu})$  telle que  $x_0 = c$  et  $\nu(x_{i+1}) = x_i$ . Si  $F$  n'apparaît dans aucune configuration de la suite  $x$ , alors pour tout  $i$ ,  $\nu(x_{i+1}) = f(x_{i+1})$  car seul le point  $i$  de la définition de  $\lambda$  est appliqué, et donc pour tout  $i$ ,  $x_i = f(x_{i+1})$ . On a donc  $c \in \Omega(f)$ .

Sinon, il existe une plus petite position  $i$  telle que  $x_i$  contient  $F$ . Par conséquent, cela signifie que le point  $i$  est utilisé dans le calcul  $x_i = \nu(x_{i+1})$ , et donc que  $x_{i+1}$  contient un sous-motif carré de taille au moins  $2|F|(l+1)$  qui est inclus dans un motif  $\hat{N}$ , où  $N$  est de taille  $l+1$  et vérifie  $\nu$ . On en déduit l'existence d'un motif de taille  $l+1$  qui vérifie  $\nu$ . Cependant, comme ce motif contient encore des motifs  $F$ , cela signifie que pour avoir généré tous ces motifs  $F$ , le point  $i$  est utilisé en chaque position d'un motif  $F$  pour le calcul  $\nu(x_{i+2}) = x_{i+1}$ . La régularité et la structure des motifs  $\hat{N}$  imposent l'existence dans  $x_{i+2}$  d'un sous-motif de taille au moins  $4|F|(l+1)$  qui est inclus dans un motif  $\hat{N}$ , où  $N$  est de taille  $2(l+1)$  et vérifie  $\nu$ . On en déduit l'existence d'un motif de taille  $2(l+1)$  qui vérifie  $\nu$ . Par récurrence immédiate, en continuant ce processus, on peut donc construire une suite de motifs de taille strictement croissante vérifiant  $\nu$ . On peut donc en déduire, par procédé d'extraction diagonale, un pavage produit par  $\nu$ , ce qui est absurde.

On en déduit que  $\nu$  produit un pavage si et seulement si  $\hat{\nu}$  n'a pas le même ensemble limite que  $c$ . Comme on peut construire  $\hat{\nu}$  récursivement en fonction de  $\nu$ , on en déduit que l'ensemble des automates cellulaires qui ont le même ensemble limite que  $f$  est many-one complet, et donc non récursif. □

Dans [Kar94], Kari a démontré un résultat similaire, mais dans lequel le nombre d'états de l'automate augmente.

Comme dans le cas des pavages, la même preuve nous permet de démontrer le résultat suivant : soit  $f$  un automate cellulaire de dimension 2 à deux états. On définit le problème suivant :

**Problème 6** (ELACN( $c$ )).

**Entrée:** Un automate cellulaire  $g$  sur  $\{0, 1\}$  de dimension 2 ;

**Question:** l'automate  $g$  a-t-il son ensemble limite inclus dans celui de  $f$ , c'est-à-dire  $\Omega(g) \subset \Omega(f)$  ?

**Théorème 15** (Variante du théorème de Rice pour les automates cellulaires).

Soit  $f$  un automate cellulaire de dimension 2, à deux états (0 et 1), dont

*l'ensemble limite n'est pas l'ensemble de toutes les configurations. Le problème  $\text{ELACN}(f)$  est indécidable.*

*Preuve.* Dans la preuve du théorème précédent, à partir de la contrainte locale  $\nu$ , on construit l'automate cellulaire  $\hat{\nu}$  telle que : si  $\nu$  produit une configuration binaire alors l'ensemble limite de  $\hat{\nu}$  est strictement plus grand que celui de  $c$ , soit  $\Omega((\nu)c) \subsetneq \Omega((\hat{\nu})c) \implies \hat{\nu} \notin \text{ELACN}(c)$ ; si  $\nu$  n'en produit pas, alors  $\hat{\nu}$  a le même ensemble limite que  $c$ , soit  $\mathcal{P}(c) = \mathcal{P}(\nu) \implies \hat{\nu} \in \text{ELACN}(c)$ . On a donc bien réduit le *domino problem* à  $\text{ELACN}(c)$ .  $\square$

Contrairement au cas des pavages, le problème suivant reste ouvert :

**Problème Ouvert 5.** *Si  $f$  est un automate cellulaire de dimension 2, le problème  $\text{ELACI}(f)$  est-il  $\Sigma_1$  ?*

De plus, là encore, ces résultats ne sont que des résultats partiels. On aimerait prouver un résultat plus fort :

**Problème Ouvert 6.** *Existe-t-il une bonne définition de "trivial" telle que pour tout ensemble  $\mathcal{E}$ , non "trivial", d'ensembles de configurations binaires, savoir si l'ensemble limite d'un automate cellulaire appartient à  $\mathcal{E}$  est indécidable ?*

Le problème suivant reste aussi ouvert :

**Problème Ouvert 7.** *Que dire du cas des automates cellulaires de dimension 1 ?*

En effet, bien que la surjectivité est décidable en dimension 1, on ne sait pas si un mot est interdit dans un ensemble limite non plein est toujours indécidable.

# Conclusion

L'objet de cette thèse est d'étudier la complexité des pavages et des automates cellulaires, d'une part à l'aide de considérations structurelles, d'autre part grâce à des notions algorithmiques.

Nos investigations de la complexité des automates cellulaires, nous ont menés à les étudier dans le cadre des systèmes dynamiques. On apporte plusieurs résultats nouveaux à l'étude des automates cellulaires au sein de la topologie de Besicovitch. La caractérisation des points fixes et périodiques que l'on a donnée témoigne du caractère intrinsèque de stabilité de ces automates. Nous pensons que ces résultats peuvent être ultérieurement améliorés en introduisant une nouvelle notion de point temporellement quasipériodique, c'est-à-dire des configurations dont l'évolution n'est pas forcément périodique, mais qui contient de vastes domaines sur lesquels on peut observer un comportement périodique. On conjecture que pour tout automate cellulaire "non trivial", l'ensemble de configurations de ce type est dense dans l'espace des phases.

Ensuite, nous introduisons la notion de "configuration de Toeplitz". On montre que cette classe de configurations, au sein de la topologie de Besicovitch, joue le même rôle que la classe des configurations spatialement périodiques pour la topologie de Cantor. Ce parallèle peut être renforcé en montrant que dans l'antécédent d'une classe contenant une configurations de Toeplitz, on trouve toujours une configuration de Toeplitz. Cependant, l'ensemble de ces configurations est assez vaste. Il est donc naturel de se demander quel serait un ensemble, à la fois facilement manipulable, et aussi minimal (du point de vue de la cardinalité) ayant les mêmes propriétés de stabilité et de densité.

Enfin, la preuve qu'il n'existe pas d'automate cellulaire transitif dans l'espace de Besicovitch a pour conséquence qu'il n'y a ni automate expansif, ni automate fortement transitif, ni automate ergodique, et à posteriori, aucun automate chaotique. On se demande donc quelles sont les notions nécessaires pour raffiner la description des automates sensibles aux conditions initiales dans la topologie de Besicovitch, et quelles peuvent être leur signification dans le contexte habituel.

De plus, dans la preuve de ce théorème, on a utilisé une technique originale dans ce domaine, inspirée de la méthode d'incompressibilité décrite dans [LV97]. On pense que bien d'autres preuves dans ce domaine peuvent tirer partie de cette technique.

La poursuite de l'étude de la complexité du comportement des automates cellulaires s'effectue par le biais de la définition d'une distance algorithmique. Cette étude se développe sur deux parties distinctes. L'une est purement topologique, et l'on y montre que l'espace est bien adapté à la modélisation par système dy-



namique discret. L'autre prouve les résultats déjà annoncé à plusieurs reprises, sur l'incapacité des automates cellulaires à transformer continûment l'information des configurations. Les résultats de la première partie sont satisfaisants, mais cependant, le problème de la complétude reste ouvert. C'est à priori un problème difficile, car il faut faire intervenir le même type d'arguments que pour la preuve, assez complexe, du lemme qui dit que si la complexité algorithmique du préfixe de taille  $n$  d'une configuration, sachant  $n$  est bornée, alors cette configuration est récursive. Grâce à cela, on pourrait utiliser de nombreux résultats classiques en topologie générale, comme, par exemple le théorème du point fixe.

Dans la seconde partie, on a montré que les automates cellulaires non surjectifs ne sont jamais continus, mais on conjecture qu'il sont continus sur leur ensemble limite, et que, de plus, ils sont l'identité sur cet ensemble. Pour cela, on manque d'un résultat profond majorant le nombre d'antécédents d'un points dans l'ensemble limite. On connaît ce résultat dans le cas où l'automate cellulaire est surjectif, mais tout autre cas demeure inconnu.

Dans la dernière partie de la thèse, on s'est concentré sur l'étude des pavages, qui sont dans certains cas considérés comme variante statique des automates cellulaires. En effet, bien des preuves difficiles sur les automates cellulaires utilisent les pavages (voir par exemple [Kar92, Kar94]). Nous nous sommes plutôt intéressés à la notion de fonction de quasipériodicité, et on a montré qu'effectivement toute fonction est capturée par un ensemble de tuile. L'étape suivante serait d'étendre cette étude, en donnant une définition de pavage *physiquement réalisable*. Ainsi, on espère mieux décrire la genèse d'un quasicristal : toute erreur sur le pavage doit pouvoir être corrigée et ne pas stopper sa croissance. Il faudrait donc introduire une notion de "couche" de tuiles, telle qu'à chaque étape, soit on peut ajouter une couche supplémentaire en respectant les règles de pavabilité, soit il suffit de remplacer la dernière couche par une autre pour ne plus être bloqué. Cette étude pourrait se révéler intéressante, car la compréhension théorique des quasicristaux est très importante. En effet, ces derniers ont de plus en plus d'applications dans la physique et la chimie.

L'étude de la complexité algorithmique des pavages mène à la recherche d'un théorème de Rice. Cependant, le résultat n'est que partiel : si l'on maintient le parallèle avec le théorème de Rice original, il faudrait non pas considérer les ensembles des tuiles qui produisent les mêmes pavages qu'un autre, mais ceux dont l'ensemble des pavages produit appartient à une certaine classe (d'ensemble de pavages) "non triviale" fixée. L'établissement d'un tel résultat est loin d'être évident. En effet, rien que la définition d'ensemble "trivial" n'est pas facile à préciser. Il existe, en effet, beaucoup de cas pour lesquels le problème est décidable. Par exemple, la classe associée au problème qui consiste à savoir si un ensemble de tuiles produit *au moins* certains pavages simples (périodiques par exemple) devrait être définie comme "triviale". Il semble donc y avoir beaucoup de classes "triviales", sans aucun point commun apparent.

En ce qui concerne les automates cellulaires, on prouve un résultat similaire sur les ensembles limites des automates, bien qu'il manque encore la preuve que le problème correspondant est  $\Sigma_1$ . De plus, là encore, l'extension à un cas plus général se heurte au problème de la définition de "trivialité". Sinon, le cas de la dimension 1 est encore à explorer, car, même si l'on sait que, dans ce cas, la surjectivité (ensemble limite plein) est décidable, on ne sait toujours pas si l'ensemble des couples  $(f, x)$  tels que  $x$  est un mot interdit de l'ensemble limite

de  $f$ , est décidable ou non.



# Bibliographie

- [AD96] C. Allauzen and B. Durand, *Appendix A : “Tiling problems”*, The classical decision problem (see [BGG96]) (1996), 407–420.
- [AP72] S. Amoroso and Y.N. Patt, *Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures*, J. Comp. Syst. Sci. **6** (1972), 448–464.
- [BCF] F. Blanchard, J. Cervelle, and E. Formenti, *Recent advances in the study of chaotic behaviour of cellular automata*.
- [Ber66] R. Berger, *The undecidability of the domino problem*, Mem. Am. Math. Soc. **66** (1966).
- [BFK99] F. Blanchard, E. Formenti, and P. Kůrka, *Cellular automata in the Cantor, Besicovitch and Weyl topological spaces*, Complex Systems **11** (1999), 107–123.
- [BGG96] E. Börger, E. Grädel, and Y. Gurevich, *The classical decision problem*, Springer-Verlag, 1996.
- [BGL<sup>+</sup>98] C. H. Bennet, P. Gács, M. Li, P. M.B. Vitányi, and W. H. Zurek, *Information distance*, IEEE Trans. Inform. Theory **44** (1998), no. 4, 1407–1423.
- [Cal94] C. Calude, *Information and Randomness*, Springer-Verlag, 1994.
- [CD00] J. Cervelle and B. Durand, *Tilings : Recursivity and regularity*, STACS’00, LNCS, vol. 1770, Springer Verlag, 2000, pp. 491–502.
- [CDF01] J. Cervelle, B. Durand, and E. Formenti, *Algorithmic information theory and cellular automata dynamics*, MFCS’01, LNCS, vol. 2136, Springer Verlag, 2001, pp. 248–259.
- [CFMM97] G. Cattaneo, E. Formenti, L. Margara, and J. Mazoyer, *A shift-invariant metric on  $S^Z$  inducing a non-trivial topology*, MFCS’97, 1997.
- [Cha66] G. J. Chaitin, *On the length of programs for computing finite binary sequences*, J. of ACM **13** (1966), 547–569.
- [CHJW00] C. Calude, P. Hertling, H. Jürgensen, and K. Weihrauch, *Randomness on full shift spaces*, Chaos, Solitons & Fractals **1** (2000), 1–13.
- [ČPY89] K. Čulik, J. Pachl, and S. Yu, *On the limit set of cellular automata*, SIAM Journal on Computing **18** (1989), 167–175.
- [Dev89] R. L. Devaney, *Introduction to chaotic dynamical systems*, second ed., Addison-Wesley, 1989.

- [DFM01] M. Delorme, E. Formenti, and J. Mazoyer, *Open problems on cellular automata*, Tech. Report RR-2000-25, Ecole Normale Supérieure de Lyon, 2001.
- [Dur98] B. Durand, *Global properties of cellular automata*, Cellular Automata and Complex Systems (E. Goles and S. Martinez, eds.), Kluwer, 1998.
- [Dur99] ———, *Tilings and quasiperiodicity*, Theo. Comp. Sci. **221** (1999), 61–75.
- [GK72] Y. Gurevich and I. Koryakov, *Remarks on Berger’s paper on the domino problem*, Siberian Math. J. **13** (1972), 319–321.
- [GS86] B. Grunbaum and G. C. Shepard, *Tilings and patterns*, V. H. Freeman Inc., 1986.
- [Kar92] J. Kari, *The nilpotency problem of one dimensional cellular automata*, SIAM Journal on Computing **21** (1992), 571–586.
- [Kar94] ———, *Rice’s theorem for the limit sets of cellular automata*, Theo. Comp. Sci. **127** (1994), 229–254.
- [Knu94] C. Knudsen, *Chaos without nonperiodicity*, American Mathematical Monthly **101** (1994), 563–565.
- [Kol65] A. N. Kolmogorov, *Three approaches to the definition of the concept “quantity of information”*, Problems of information transmission **1** (1965), 3–11.
- [Kûr97] P. Kûrka, *Languages, equicontinuity and attractors in cellular automata*, Erg. Th. & Dyn. Sys. **217** (1997), 417–433.
- [LV97] M. Li and P. Vitányi, *An introduction to Kolmogorov complexity and its applications*, second ed., Springer-Verlag, 1997.
- [Man98] G. Manzini, *Characterization of sensitive linear cellular automata with respect to the counting distance*, MFCS’98, 1998.
- [MK76] A. Maruoka and M. Kimura, *Conditions for injectivity of global maps for tessellation automata*, Information & Control **32** (1976), 158–162.
- [ML66] P. Martin-Löf, *The definition of a random sequence*, Information & Control **9** (1966), 602–619.
- [Moo63] E. F. Moore, *Machine models of self-reproduction*, Proc. Symp. Appl. Math., AMS Rep. **14** (1963), 17–34.
- [Myh63] J. Myhill, *The converse of Moore’s Garden of Eden theorem*, Proc. Amer. Math. Soc. **14** (1963), 685–686.
- [Nas95] M. Nasu, *Textile systems for endomorphisms and automorphisms of the shift*, Memoirs of the AMS **546** (1995).
- [Rob71] R.M. Robinson, *Undecidability and nonperiodicity for tilings of the plane*, Inventiones Mathematicae **12** (1971), 177–209.
- [Sol64] R. J. Solomonoff, *A formal theory of inductive inference, part 1 and part 2*, Inform. Contr. **7** (1964), 1–22, 224–254.
- [Sut91] K. Sutner, *De Bruijn graphs and linear cellular automata*, Complex Systems **5** (1991), 19–30.

- [US96] V. A. Uspensky and A. Kh. Shen, *Relations between varieties of Kolmogorov complexities*, Math. Syst. Theory **29** (1996), no. 3, 270–291.
- [Wol86] S. Wolfram, *Theory and applications of cellular automata*, World Scientific, Singapore, 1986.

# Index

- $\underline{0}$ , 13
- $\underline{1}$ , 13
- $2^x$ , 45
- $\doteq$ , 45
- $\dot{=}$ , 25
- $\llbracket a, b \rrbracket$ , 12
- $\text{BO}(x, r)$ , 17
- $d_{\mathcal{B}}$ , 24
- $d_{\mathcal{X}}$ , 45
- $\text{Dom}(M)$ , 66
- $\Delta(u, v)$ , 24
- $f$ , 26, 50
- $K_{\varphi}(x|y)$ , 40
- $K(x)$ , 41
- $K(x|y)$ , 41
- $\Omega(f)$ , 53
- $\Omega^n(f)$ , 53
- $|M|$ , 66
- $\mathcal{M}_x(n)$ , 13
- $\mathcal{O}_f(x)$ , 13
- $\mathcal{P}(c)$ , 72
- $p|u|_p$ , 26
- $\mathcal{Q}_P$ , 67
- $\sigma$ , voir *décalage*
- $\mathcal{T}$ , 34
- $[w]$ , 14
- $X$ , 12, 13
- $x/A$ , 43
- $x//_A^n$ , 43
- $x_{a:b}$ , 13
- $\dot{X}$ , 25, 45
- $x_i$ , 13
- $\langle x, y \rangle$ , 40
- automates cellulaires, 12
  - configuration, 12, 14
  - diagramme espace-temps, 14
  - dimension, 12
  - états, 12
  - règle locale, 12
  - voisinage, 12
- $c$ -incompressible, 41
- classes de K urka, 15
  - classe I, 17
  - classe II, 18
  - classe III, 18
  - classe IV, 18
  - exemples, 20
- codage autod elimit e, 42
- compatibilit e, 26, 50
- complexit e algorithmique, 40
  - conditionnelle, 41
- complexit e de Kolmogorov, 40
- configuration
  - configuration finie, 13
  - faiblement incompressible, 42
  - log*-incompressible, 42
  - sous-motif, 67
  - spatialement p eriodique, 13
  - spatialement p eriodiques, 33
- configurations
  - de Toeplitz, 34
- constante d'expansivit e, 18
- constante de sensibilit e, 18
- construction autosimilaire, 63
- contrainte locale, 58
- cylindre, 14
- d ecalage, 20
- domino problem, 62, 66, 68, 74, 76
- ensemble limite, 53
-  equicontinuit e, 15
- expansivit e, 18
- extraction diagonale, 59
- fonction constructible en temps, 68
- formes rationnelles, 58
- forte transitivit e, 19
- mode de description, 40
  - additivement optimal, 40
- mot bloquant, 16

- motif, 66
  - interdit, 76
- pavabilité, 62, 66
  - contrainte locale, 58
  - formes rationnelles, 58
  - pavabilité périodique, 63, 66
  - tuiles colorées, 56
  - tuiles fléchées, 57
- pavage de Robinson, 64
- pavage périodique, 62
  - autosimilarité, 63
- problème
  - ELACI( $f$ ), 76
  - ELACN( $c$ ), 78
  - PCLI( $c$ ), 72
  - PCLN( $c$ ), 75
- pseudo-distance, 24
- pseudo-distance algorithmique, 45
- quasipériodicité, 66
  - fonction, 66, 67
  - pavages autosimilaires, 66, 67
- sensibilité, 18
- sous-shift, 76
- topologie de Besicovitch, 24
  - pseudo-distance, 24
- topologie de Cantor, 14
  - distance, 14
- transitivité, 19
- tuiles colorées, 56
- tuiles fléchées, 57



# Complexité structurelle et algorithmique des pavages et des automates cellulaires

## RÉSUMÉ

Ce travail de thèse étudie la complexité des pavages et des automates cellulaires.

L'analyse débute par des considérations structurelles : la quasipériodicité des pavages. À tout ensemble de tuiles qui pave le plan, on associe une fonction de quasipériodicité qui quantifie sa complexité. Tout d'abord, on montre que toute fonction "raisonnable" peut être capturée par un ensemble de tuiles et qu'il existe des pavages dont la fonction de quasipériodicité croît plus rapidement que n'importe quelle fonction récursive. Ensuite, on démontre un théorème de Rice pour les pavages : l'ensemble des ensembles de tuiles qui admettent les mêmes pavages qu'un autre fixé est indécidable et récursivement énumérable. Enfin, on transpose notre résultat dans le contexte des automates cellulaires.

La seconde partie de notre travail concerne l'étude des automates cellulaires sous l'angle des systèmes dynamiques, et plus particulièrement des automates chaotiques. Les définitions usuelles classifiant les automates chaotiques ne sont pas satisfaisantes. Pour palier ce problème, on utilise deux nouvelles topologies. La première est dite de Besicovitch, et permet de supprimer la prédominance du motif central lors de l'étude de l'évolution de l'automate. On apporte plusieurs résultats, les premiers indiquant que notre nouvel espace de travail est acceptable à l'étude des automates cellulaires, en tant que systèmes dynamiques ; les seconds montrent que la notion de chaos subsiste, grâce à la définition de sensibilité aux conditions initiales, mais que les classes plus chaotiques sont vides.

La seconde topologie employée est définie à l'aide de la complexité algorithmique. Le but de cette approche est d'avoir une distance qui traduit la facilité à calculer un élément à l'aide de l'autre. Nos résultats complètent les précédents. Ils attestent de manière formelle que les automates cellulaires ne peuvent pas modifier continûment l'information contenue dans une configuration, et surtout qu'ils sont incapables d'en créer.

---

## Structural and algorithmic complexity of tilings and cellular automata

### ABSTRACT

This work deals with the complexity of tilings and cellular automata.

The analysis begins with structural considerations : quasiperiodicity of tiling. Each tiling is associated with a quasi-periodicity function that quantifies the degree of complexity. We prove that almost every "reasonable" function can be obtained as quasi-periodicity function of a given set of tiles. Moreover quasi-periodicity function grows faster than any recursive function. This results are completed by a Rice theorem for tilings : the class of set of tiles that admits the same tiling is undecidable and recursively enumerable. Finally a similar result is proved for cellular automata.

In the second part of the work we focus on the study of cellular automata from a dynamical systems point of view with emphasis on chaotic behavior. Actually, present definitions of chaotic behavior are not completely satisfying. For this reason two new topologies are introduced and studied. As far as the first topology, called Besicovitch topology, we give several new results culminating with the proof that there are no transitive cellular automata. This fact proves that cellular automata have simple dynamics. This fact is illustrated even further by the results obtained when cellular automata phase space is endowed with the second topology. In fact this second topology is defined via Kolmogorov complexity. Here two points are near if it is easy to find an algorithm to transform the first point in the other and vice-versa. In this new context we proved that cellular automata are not able to modify continuously the information content of the initial configuration and by no means they can add randomness to configurations.

---

**Mots clef :** Pavages, Automates cellulaires, Systèmes dynamiques discrets, Chaos déterministe, Complexité algorithmique, Décidabilité.