HAL open science

Automates cellulaires : structures

Nicolas Ollinger

To cite this version:

Nicolas Ollinger. Automates cellulaires : structures. Autre [cs.OH]. Ecole normale supérieure de lyon - ENS LYON, 2002. Français. NNT : . tel-00007765

HAL Id: tel-00007765

https://theses.hal.science/tel-00007765v1

Submitted on 15 Dec 2004

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.

N° d'ordre : 241
N° attribué par la bibliothèque : 02ENSL0241

ÉCOLE NORMALE SUPÉRIEURE DE LYON
Laboratoire de l'Informatique du Parallélisme

THÈSE
pour l'obtention du grade de
Docteur de l'École normale supérieure de Lyon

spécialité : Informatique

au titre de l'école doctorale de : Mathématiques et Informatique fondamentale

présentée et soutenue publiquement le 19/12/2002
par Monsieur Nicolas OLLINGER

Automates cellulaires : structures

Directeurs de thèse :	DELORME Marianne	
	MAZOYER Jacques	
Après avis de :	Monsieur François BLANCHARD,	Membre/Rapporteur
	Monsieur Géraud SÉNIZERGUES,	Membre/Rapporteur

Devant la Commission d'examen formée de :

Monsieur François BLANCHARD,	Membre/Rapporteur
Madame Marianne DELORME,	Membre/Directeur de thèse
Monsieur Jarkko KARI,	Membre
Monsieur Jacques MAZOYER,	Membre/Directeur de thèse
Monsieur Dominique PERRIN,	Membre
Monsieur Géraud SÉNIZERGUES,	Membre/Rapporteur

REMERCIEMENTS

Une thèse se construit sur des rencontres et des échanges. Je voudrais remercier ici toutes les personnes qui ont contribué — directement ou indirectement — à l'aboutissement de mes travaux de thèse, concrétisé par le présent mémoire.

Je remercie en premier lieu mes directeurs de thèse, Marianne Delorme et Jacques Mazoyer qui ont su m'accompagner et me conseiller durant ces années de thèse. Leur complémentarité sur le plan scientifique m'a offert un espace de grande liberté dans un cadre sécurisant de rigueur. Leur disponibilité, leur patience et leur complicité ont permis un travail efficace mais aussi ont su répondre à mes questions, apaiser mes craintes et participer à ma découverte du métier d'enseignant-chercheur, sous tous ses aspects.

Je remercie François Blanchard et Géraud Sénizergues qui ont accepté de rapporter sur cette thèse. Leurs lectures minutieuses de ce mémoire et leurs profils si différents ont donné lieu à des rapports complémentaires apportant de nouveaux regards sur mes travaux. Je remercie aussi Jarkko Kari et Dominique Perrin qui se sont joints aux précédents pour compléter mon jury de thèse.

Je remercie les membres de l'équipe de recherche MC^2, du LIP, qui m'ont accueilli durant cette thèse. Je remercie en particulier mes co-bureaux successifs, pour leurs discussions — scientifiques, administratives et autres — autour d'un café ou d'un tableau : Bruno Martin, Hervé Fournier, Christophe Papazian et Grégory Lafitte.

Je remercie toutes les personnes rencontrées au cours des nombreux voyages réalisés durant cette thèse. Que ce soit à Giens, Prague, Malága, Santiago ou Riga, à l'occasion de conférences, d'ateliers de travail ou d'écoles, pour de courtes discussions ou de plus longues séances devant un tableau, ces échanges ont beaucoup influencé mon travail. Qu'il me soit permis de remercier plus particulièrement Maurice Nivat, Ivan Rapaport, Guillaume Theyssier, Bruno Durand, Enrico Formenti, Georges Varouchas, Katsunobu Imai, Manfred Kudlek, ainsi que Véronique Terrier.

Je remercie Juhani Karhumäki et Jarkko Kari pour m'avoir accueilli lors de deux longs séjours à l'université de Turku, en Finlande, tant pour la richesse de leurs apports scientifiques que pour leur grande hospitalité.

Je remercie les différentes personnes, étudiants et enseignants, rencontrées au cours de ma scolarité — au sein du magistère MIM à l'ENS Lyon, puis à l'ENS de Cachan, au sein du DEA Algorithmique — et qui ont influencé mes choix en matière de thème de recherche. Je remercie en particulier ZAG, JC, Jym, Fred, Laurent, Aymeric et Marie.

Je remercie enfin ma famille pour sa présence et son soutien tout au long de ma scolarité.

Je ne saurais clore cette page sans remercier Alexandra pour sa présence, son attention et sa patience... Merci.

TABLE DES MATIÈRES

Notations **9**

Introduction **11**

1 Automates Cellulaires **19**
1.1 Automates cellulaires . . . 19
1.1.1 L'objet et sa définition . . . 20
1.1.2 Discussion et propriétés . . . 20
1.1.3 Une caractérisation par clôture . . . 30
1.2 Classifications . . . 33
1.2.1 Genèse . . . 33
1.2.2 Classification, calculabilité et complexité . . . 45

2 Groupages **47**
2.1 Vers une extension du groupage . . . 48
2.1.1 Algorithmique . . . 48
2.1.2 Automates cellulaires unidirectionnels . . . 48
2.1.3 Automates cellulaires nilpotents . . . 49
2.1.4 Universalité intrinsèque . . . 49
2.1.5 Synthèse . . . 50
2.2 Interlude formel . . . 51
2.2.1 Axiomatisation . . . 51
2.2.2 Préordre . . . 54
2.2.3 Universalités . . . 55
2.3 Première esquisse du groupage . . . 56
2.3.1 Transformations géométriques . . . 57
2.3.2 Axiomatisation . . . 67
2.4 Une extension du groupage carré . . . 71
2.4.1 Définitions . . . 71
2.4.2 Espace des phases . . . 72
2.4.3 Exploration . . . 73
2.4.4 Familles classiques d'automates cellulaires . . . 80
2.4.5 Structure . . . 81
2.5 Un groupage fortement structuré . . . 83
2.5.1 Transformations géométriques généralisées . . . 83
2.5.2 Axiomatisation . . . 85
2.5.3 Exploration . . . 85

3 Universalités **87**
3.1 Différentes notions d'universalité . . . 87
3.1.1 Universalités . . . 87
3.1.2 Un résultat étonnant . . . 88

3.1.3 Prototypes ayant de bonnes propriétés 89
3.1.4 Minimisation 90
3.2 Indécidabilité de l'universalité intrinsèque 91
3.2.1 Un automate cellulaire intrinsèquement universel 91
3.2.2 Automates cellulaires chaudières 92
3.2.3 Résultat 93
3.3 Un automate universel bilinéaire à deux états 94
3.4 Un automate universel à 6 états 97
3.4.1 Un automate cellulaire universel simple à 8 états 97
3.4.2 Affinement du nombre d'états requis 102

4 Pour aller plus loin 107

A Groupage carré 109
A.1 Définitions 109
A.2 Propriétés élémentaires 110
A.3 Le bas de l'ordre 111
A.3.1 Automates cellulaires dynamiquement simples 111
A.3.2 Automates cellulaires algébriquement simples 113
A.4 Chaînes infinies bornées 114
A.5 Chaînes infinies non-bornées 115

TABLE DES FIGURES

1.1 Exemples de diagrammes espace-temps partiels 21
1.2 Quelques voisinages classiques 23
1.3 Propriétés de la règle globale de transition et familles de configurations . . 28
1.4 Diagrammes espace-temps des 256 automates cellulaire élémentaires 34
1.5 Diagrammes espace-temps d'automates cellulaires chaotiques (1/2) 37
1.5 Diagrammes espace-temps d'automates cellulaires chaotiques (2/2) 38
1.6 Diagrammes espace-temps d'automates cellulaires complexes (1/2) 40
1.6 Diagrammes espace-temps d'automates cellulaires complexes (2/2) 41
1.7 Automates cellulaires complexes : fonds, particules et collisions 42
1.8 Automates cellulaires complexes : abstraction des collisions 42
1.9 Opération de zoom arrière 42
1.10 Opération de groupage 42
1.11 Diagrammes espace-temps d'automates cellulaires complexes groupés (1/2) 43
1.11 Diagrammes espace-temps d'automates cellulaires complexes groupés (2/2) 44

2.1 Transformation géométrique de P. C. Fischer 48
2.2 Transformation géométrique de C. Choffrut et Čulik II 49
2.3 Transformation géométrique de O. H. Ibarra *et al.* 49
2.4 Transformation géométrique et nilpotence 49
2.5 Transformation géométrique et universalité intrinsèque 50
2.6 Transformation $\langle m, n, k\rangle$ d'un diagramme espace-temps 51
2.7 Simulation entre objets dans le groupage abstrait 52
2.8 Exemple de transformation géométrique : $\left(3, \mathbf{\Lambda}^{\langle 3,4,1\rangle}\right)$ 58
2.9 Opérateurs géométriques 59
2.10 Réseau associé à une famille de vecteurs de $\mathbb{Z}^2$ et pavage 60
2.11 Composition de plongements de $\mathbb{Z}^2$ dans lui-même 61
2.12 Contre-exemple pour l'axiome de **Proximité** 68
2.13 Espace des phases d'un automate cellulaire 73
2.14 Rayon de voisinage et transformations géométriques dans $\mathbb{Z}^2$ 74
2.15 Espace des phases de $\boldsymbol{\xi}^{\mathcal{P}}_{m,n}$ 75
2.16 Comparaison des $\boldsymbol{\xi}_{m,n}$ 76

3.1 Triangle de Pascal modulo deux engendré par $\mathcal{B}$ 94
3.2 Principe de codage d'une cellule 95
3.3 Une enveloppe de cellule $(1, 3)$ 96
3.4 Règle locale d'un automate cellulaire et sa décomposition 98
3.5 Fonctions booléennes et représentation par des circuits à niveaux 98
3.6 Codage de cellules dans le simulateur booléen d'automates cellulaires . . . 99
3.7 Trois types d'états pour déplacer l'information 100
3.8 Codage d'un automate cellulaire dans l'automate universel à 8 états 101
3.9 Codage d'un automate cellulaire dans l'automate universel à 7 états 102
3.10 Lieux du calcul dans les automates cellulaires universels 103
3.11 Codage d'une cellule, indicé par les opérations des signaux poubelle 103

3.12 Opérations de nettoyage des signaux poubelle 104
3.13 Codage d'un automate cellulaire dans l'automate universel à 6 états 104

A.1 Diagramme de dépendances d'un automate cellulaire 109
A.2 Le bas de l'ordre $\leqslant_{\square}$: classes et représentants 114
A.3 Chaînes infinies bornées pour l'ordre $\leqslant_{\square}$ 115
A.4 Une chaîne infinie non-bornée pour l'ordre $\leqslant_{\square}$ 116

NOTATIONS

$\mathbb{N}$	entiers naturels
$\mathbb{Z}$	entiers relatifs
d	dimension de l'espace
Q	ensemble fini d'états
C, D	configurations
p	position de $\mathbb{Z}^d$
C_p	état de la cellule de position p
r	rayon de voisinage
$[\![a, b]\!]$	ensemble des entiers de l'intervalle $[a, b]$
$\|p\|_1$	norme euclidienne
$\|p\|_\infty$	norme infinie
$\mathcal{A}$, $\mathcal{B}$	automates cellulaires
M	motif fini d'états
V	voisinage
ν	taille du voisinage
v_i	vecteur de voisinage
$\|V\|$	cardinal de V
δ	règle locale de transition
G	règle globale de transition
$Q^{\mathbb{Z}^d}$	ensemble des configurations
Δ	évolution
$\operatorname{Im} G$	image de l'application G
$\pi_i(x)$	i-ème composante du uple x
$\|x\|$	nombre d'éléments du uple x

INTRODUCTION

« Nous ne sommes que des grains de sable
mais nous sommes ensemble.
Nous sommes comme les grains de sable
sur la plage, mais sans les grains de sable
la plage n'existerait pas. »

poème en langue Yamato

La recherche se trouve actuellement confrontée, dans différentes disciplines et domaines d'étude, à la notion — encore informelle mais déjà unificatrice — de système complexe. Une problématique majeure dans l'étude de tels systèmes consiste à comprendre comment un ensemble d'objets interagissant selon des règles locales déterminées peut engendrer un comportement global complexe, difficile à comprendre au simple vu des règles locales. En d'autres termes, cette problématique consiste à expliquer pourquoi et comment des systèmes peuvent avoir un comportement microscopique facile à décrire mais un comportement macroscopique beaucoup plus compliqué.

Parmi les nombreux modèles de systèmes complexes existant, les automates cellulaires se caractérisent par leur simplicité en tant qu'objets mathématiques : une grille régulière et infinie d'entités simples toutes identiques, chacune dans un état pris parmi un ensemble fini, évolue de manière déterministe et synchrone via une règle d'évolution définie localement par une règle de changement d'état appliquée uniformément. Les automates cellulaires fournissent ainsi une sorte de modèle épuré, simplifié mais non simpliste, dans lequel étudier la problématique des interactions entre comportement local et global. En particulier, le choix de travailler sur une collection infinie d'objets interagissant uniformément, à la fois en espace, de par leur agencement régulier, et en temps, de par le caractère synchrone du modèle, permet de contourner les redoutables problèmes de nature combinatoire qui apparaissent lorsqu'on considère des modèles discrets finis. De nombreux travaux ont été entrepris pour améliorer la compréhension des automates cellulaires et de leur dynamique, selon des points de vue différents et avec des outils très variés provenant tantôt de la théorie classique des systèmes dynamiques, tantôt de l'algèbre ou de la topologie, tantôt de la physique statistique, tantôt de la biologie, tantôt de l'informatique théorique.

Travaux antérieurs

À la suite des travaux pionniers de J. von Neumann [80] et S. Ulam d'une part et K. Zuse [83] d'autre part, motivés par une volonté de modélisation de phénomènes complexes telles la notion d'auto-reproduction chez les organismes vivants pour J. von Neumann et la modélisation des lois de la physique pour K. Zuse, une étude systématique des automates cellulaires a été entreprise.

Ainsi, les propriétés des règles globales ont été analysées du point de vue algébrique dans le prolongement des travaux de E. F. Moore [16] et J. Myhill [17]. D'autres paradigmes importants comme la notion de signal ont été mis à jour par une étude algorithmique amorcée par P. C. Fischer [18]. Une importante caractérisation topologique des

automates cellulaires a été proposée par G. Hedlund [19] qui permet de relier les automates cellulaires aux outils classiques de la théorie des systèmes dynamiques. Ainsi, la notion d'ensemble limite, l'ensemble des comportements asymptotiques possibles d'un automate cellulaire, a été étudiée par K. Čulik II, J. Pachl et S. Yu [38]. Plus récemment, les problèmes de décidabilité liés à ces propriétés ont été analysés, comme les problèmes de la réversibilité en dimension 2 ou de la nilpotence en dimension 1 étudiés par J. Kari [44]. Les aspects calculatoires, présents dans ces problèmes de décision, se retrouvent aussi dans les évolutions des automates cellulaires. Déjà J. von Neumann propose d'introduire les calculs d'une machine de Turing dans l'évolution d'un automate cellulaire, ce qui amène à la notion d'automate cellulaire Turing universel ou universel pour le calcul. Mais J. von Neumann laisse supposer une autre forme d'universalité, présente aussi dans les travaux de E. R. Banks [20] ou de J. Albert et K. Čulik II [34]. Il s'agit là d'une forme d'universalité intrinsèque aux automates cellulaires : un automate cellulaire est intrinsèquement universel s'il est possible de trouver dans ses évolutions, à un codage régulier près, toutes les évolutions possibles des automates cellulaires. Une première formalisation de cette notion est proposée par B. Martin [54].

Ces vingt dernières années, une nouvelle approche de la problématique est apparue dans le domaine à travers la recherche de classifications pertinentes des automates cellulaires. Dans ses travaux pionniers mais hélas informels, S. Wolfram [29] a proposé, à partir de l'observation de diagrammes espace-temps d'automates cellulaires, de distinguer 4 comportements type. Les deux dernières classes de S. Wolfram correspondent aux automates cellulaires les plus complexes : les automates cellulaires de classe 3, dits « chaotiques », dont le comportement semble aléatoire, et les automates cellulaires de classe 4, pour lesquels émergent des structures, des particules, qui interagissent de manière compliquée.

De nombreuses propositions de classifications formelles reposant sur des bases mathématiques solides ont été faites pour essayer de mieux cerner les comportements de type classe 3 — en particulier des classifications reposant sur des propriétés topologiques ou encore de mesure comme celle de B. Martin II [13]. Cependant, peu de travaux ont été réalisés pour aborder les automates cellulaires de classe 4.

Dans ce cadre, I. Rapaport [12] a proposé une classification reposant sur les notions de sous-automate et de groupage et induisant une relation de préordre sur l'ensemble des automates cellulaires. Un sous-automate d'un automate cellulaire est une restriction des états pris par les cellules de cet automate cellulaire à un sous-ensemble d'états stable par application de la règle de transition. Un groupé d'un automate cellulaire est un automate cellulaire dont les diagrammes espace-temps peuvent être obtenus par une opération de « zoom arrière sans perte » appliquée aux diagrammes espace-temps de l'automate cellulaire d'origine. Cette classification par groupage se distingue des autres sur plusieurs plans. Tout d'abord, la classification obtenue possède un nombre infini de classes. D'autre part, la classification est de nature algébrique et semble adaptée à l'étude de la classe 4.

Nos travaux autour des automates cellulaires, de leur classification et du groupage en particulier partent de certains constats sur le groupage :

- la nature algébrique de cet outil en fait un outil puissant : on peut espérer utiliser des propriétés structurelles pour démontrer des résultats non triviaux sur les automates cellulaires ;
- le groupage est aussi de nature géométrique si on observe les diagrammes espace-temps, il faut comprendre cette intuition géométrique pour progresser ;

– certains choix faits dans la définition du groupage l'empêchent de prendre en compte certaines notions intéressantes comme l'universalité intrinsèque, mais la démarche est pertinente : il faut adapter le groupage.

Structure des automates cellulaires

Une première partie de nos travaux a consisté à réétudier la structure interne des automates cellulaires, ce qui nous a permis de mettre en évidence une nouvelle représentation, une sorte de forme normale pour ces objets.

L'ensemble des automates cellulaires est la clôture d'une famille d'automates cellulaires très simples par trois opérations. Les automates cellulaires simples en question sont les automates cellulaires autarciques, dans lesquels chaque cellule du réseau agit indépendamment des autres, ainsi que les automates cellulaires de translation, dans lesquels les états sont translatés uniformément à chaque temps. Les trois opérations de clôture mises en évidence sont la composition des règles globales de transition, leur produit cartésien et l'opération de passage au sous-automate. Si les deux premières opérations sont classiques, cette caractérisation met en avant la troisième qui, si elle apparaît naturellement dans le cadre du groupage, est plus originale. De plus, la restriction de l'ensemble de départ aux automates cellulaires autarciques réversibles engendre l'ensemble de tous les automates cellulaires réversibles, donnant ainsi une manière élégante d'engendrer les automates cellulaires réversibles.

Ces deux caractérisations mettent en avant un lien, qui reste à étudier, entre la complexité d'un automate cellulaire et l'ensemble des sous-automates potentiels apparaissant dans les itérés de l'automate cellulaire d'origine. Elles rendent également compte assez naturellement de la « nature » du calcul intrinsèque aux automates cellulaires : l'information est déplacée (par les translations) et modifié en place (par les automates cellulaires autarciques), l'automate cellulaire final n'étant qu'un produit d'une séquence de tels comportements, projeté par l'opération sous-automate sur un sous-ensemble de configurations.

Transformations géométriques

Forts de cette caractérisation qui motive l'utilisation de la relation sous-automate pour définir une relation de comparaison entre automates cellulaires, nous nous sommes intéressés à un autre aspect important du groupage : les transformations géométriques de diagrammes espace-temps.

Après avoir observé l'action de l'opération de groupage d'I. Rapaport sur les diagrammes espace-temps, nous avons étudié les transformations géométriques sur les diagrammes espace-temps des automates cellulaires utilisées dans la littérature. Ces observations nous ont amenés à distinguer trois transformations « naturelles » et leurs compositions : le découpage spatial, le découpage temporel et les translations. Ces opérations sont agréables à manipuler car elles s'expriment algébriquement sur les règles locales de transition, algébriquement sur les règles globales de transition, géométriquement sur les diagrammes espace-temps, voire algorithmiquement si on les compare aux opérations couramment utilisées dans les travaux algorithmiques sur automates cellulaires. Dans ce cadre, les transformations utilisées pour réaliser l'opération de groupage d'I. Rapaport

peuvent être vues comme le couplage d'un découpage spatial et d'un découpage temporel que l'on contraint à une certaine régularité.

Cette première approche nous a amenés à formaliser la notion de transformations « géométriques » sympathiques, celles qui sont intéressantes dans le cadre du groupage : les transformations définies par le regroupement de cellules des diagrammes espace-temps qui définissent complètement pour tout automate cellulaire un unique automate cellulaire transformé.

Nous avons alors démontré, la démonstration reposant sur la grande régularité des diagrammes espace-temps des automates cellulaires, que seules les compositions des transformations de découpage spatial, découpage temporel et translation sont des transformations sympathiques, ce qui fournit, ici aussi, une sorte de forme normale, cette fois-ci pour les transformations de diagrammes espace-temps.

Structure de l'ensemble des automates cellulaires

Munis de ces deux outils, la relation sous-automate et les transformations « géométriques » sympathiques, nous nous sommes intéressés à définir une extension du groupage d'I. Rapaport qui prenne en compte des phénomènes tels l'universalité intrinsèque.

Plutôt que de reprendre la même démarche pour chaque extension proposée, nous avons établi un cadre formel d'étude des groupages par le biais d'une théorie logique du premier ordre et d'une technique de démonstration géométrique. Grâce à cette théorie logique, l'introduction de nouvelles formes de groupage est simplifiée, il suffit de vérifier une famille d'axiomes pour obtenir immédiatement un certain nombre de résultats — dont le plus important : la relation de groupage obtenue est une relation de préordre. Cette structure sur l'ensemble des automates cellulaires est très importante car plus la structure est forte, plus on peut espérer progresser dans l'étude des objets.

Nous avons alors démontré que l'ensemble des transformations sympathiques ne permet pas d'obtenir une structure de préordre. Aussi, nous avons introduit une première extension du groupage reposant sur une sous-famille des transformations sympathiques. Grâce à de nouveaux outils, comme l'espace des phases, nous avons montré que le nouveau groupage restait compatible dans une certaine mesure avec celui introduit par I. Rapaport et que de nombreux résultats y étaient préservés et se démontraient aisément. D'autre part, ce premier groupage répond à nos attentes : en plus de pouvoir caractériser facilement le bas de la classification, il existe une classe maximale correspondant aux automates cellulaires intrinsèquement universels et on peut ainsi étudier la notion d'automate cellulaire de complexité maximale. Certaines familles naturelles d'automates cellulaires, comme les automates cellulaires réversibles, périodiques, nilpotents, se formalisent naturellement dans cette classification. Nous avons démontré que la relation de simulation introduite est indécidable.

Une analyse des classes d'équivalence des produits cartésiens d'automates cellulaires de translations nous a permis de montrer que la structure de l'ordre induit est assez faible : il ne s'agit pas d'un demi-treillis.

Le produit cartésien fournissant une notion de maximum local raisonnable, nous avons proposé une extension de ce groupage, fortement structurée en un demi-treillis par l'opération de produit cartésien d'automates, en quotientant la relation de simulation précédente

par l'opération de produit cartésien. Dans ce nouveau groupage, la plupart des résultats précédents sont conservés. De plus, les idéaux de ce demi-treillis capturent des familles classiques d'automates cellulaires tels les automates cellulaires surjectifs. Nous avons pu alors avec succès relier des propriétés de nature algébrique — le caractère principal de certains idéaux — à des problèmes de décision — l'appartenance à ces idéaux — par exemple dans le cas de la surjectivité ou de l'universalité intrinsèque.

Universalité intrinsèque

De même que le bas de l'ordre correspond à des familles d'automates cellulaires facilement définissables dans le cas du groupage d'I. Rapaport, le haut de l'ordre devient à son tour accessible dans nos extensions. Nous nous sommes intéressés à son étude ainsi qu'à la notion d'universalité intrinsèque.

Ainsi, nous avons montré que ces extensions du groupage permettent de définir formellement la notion d'automate cellulaire intrinsèquement universel et de démontrer la non universalité intrinsèque de certains automates cellulaires — ce qui est impossible avec la notion d'automate cellulaire universel pour le calcul.

Nous avons démontré l'indécidabilité de l'universalité intrinsèque. Cette démonstration possède un double intérêt. D'une part, la technique de preuve est originale, utilisant une méthode de propagation d'énergie, et fournit une première démonstration d'indécidabilité d'une propriété qui ne soit ni asymptotique ni résultant d'un unique pas de calcul. D'autre part, ce résultat nous a permis de démontrer que le haut de l'ordre n'est occupé que par la classe d'équivalence des automates cellulaires intrinsèquement universels : il n'y a pas de notion d'automate cellulaire « presque » intrinsèquement universel. Les automates cellulaires intrinsèquement universels étant aussi universels pour le calcul, pour toute définition raisonnable de cette notion, et sachant qu'il existe des automates cellulaires universels pour le calcul qui ne sont pas intrinsèquement universels, il existe des automates cellulaires universels pour le calcul situés à une « distance » infinie des automates cellulaires intrinsèquement universels.

Enfin, par le biais d'un résultat démontré par I. Rapaport, nous avons mis en avant un résultat pour le moins étonnant : il n'existe pas d'automate cellulaire intrinsèquement universel qui effectue ses calculs en temps réel, le temps minimal de propagation de l'information, mais il en existe qui approchent le temps réel d'aussi près que l'on veut par un automate cellulaire intrinsèquement universel calculant en temps linéaire. Il sera sans doute intéressant d'étudier les liens entre ce résultat et un ancien problème ouvert concernant la reconnaissance de langages sur automates cellulaires : comparer les puissances de reconnaissance en temps réel et en temps linéaire.

Algorithmique contrainte

Une dernière partie de nos travaux consiste en une étude algorithmique des automates cellulaires intrinsèquement universels. Nous nous sommes intéressés à la construction d'automates cellulaires intrinsèquement universels les plus simples possible pour différents critères de simplicité, tel le nombre d'états pour un automate cellulaire à voisinage fixé. Nous avons amélioré les précédentes constructions d'automates cellulaires universels bilinéaires

en ramenant le nombre d'états à 2 grâce à une méthode originale de codage de l'information non pas dans les états des cellules mais dans la distance entre deux cellules de même état. De même, nous avons construit un automate cellulaire intrinsèquement universel de dimension 1 à 6 états, le plus petit connu à ce jour, grâce à un codage optimisé par une étude fine du croisement de l'information au sein d'un réseau booléen. La pratique de ce type d'algorithmique contrainte nous a donc permis d'exhiber de nouvelles méthodes de codage de l'information en repoussant la complexité de la règle locale vers des interactions locales de paquets de cellules.

Poursuite des travaux

Ces travaux mettent en avant certaines questions qui nous semblent intéressantes à étudier pour poursuivre la compréhension des automates cellulaires en tant que « systèmes complexes » :

1. Le groupage et ses extensions nous ont permis de caractériser les automates cellulaires aux comportements les plus simples, les automates cellulaires situés au bas, et les plus compliqués, les automates cellulaires situés en haut de l'ordre induit par le groupage. La complexité maximale des automates cellulaires intrinsèquement universels est obtenue en combinant tous les comportements possibles à l'intérieur d'un unique automate cellulaire. Il s'agit de la manière la plus simple d'introduire de la complexité. Quels types de comportements exhibent alors, ou n'exhibent pas, les automates cellulaires intermédiaires ?

2. Cette structure sur les automates cellulaires peut être utilisée pour définir une distance sur les automates cellulaires de la manière suivante. Deux automates cellulaires sont à distance 0 s'ils appartiennent à une même classe d'équivalence. Deux automates cellulaires sont à distance 1 s'ils appartiennent à deux classes d'équivalence comparables telles qu'il n'existe pas de classe d'équivalence intermédiaire. Pour terminer la définition de cette distance, on prend pour distance entre deux automates cellulaires quelconques la distance maximale qui satisfasse l'inégalité triangulaire. Cette distance correspond donc à la distance non orientée dans le graphe du pré-ordre simplifié en retirant les arcs de transitivité. En particulier, les automates cellulaires intrinsèquement universels sont — grâce à nos résultats sur le haut de l'ordre — placés à distance infinie des autres. Quels types d'ordinaux rencontre-t-on dans cette distance et comment s'interprètent-ils en termes structurels sur les automates cellulaires ?

3. Le théorème de Rice utilisé en théorie de la calculabilité possède diverses extensions, y compris dans le monde des automates cellulaires. Il semble raisonnable que l'appartenance aux classes d'équivalence non triviales du groupage ainsi qu'à ses idéaux soit indécidable. Existe-t-il un analogue du théorème de Rice qui l'affirme ?

4. Partant du pré-ordre introduit par I. Rapaport, nous avons proposé différents raffinements produisant deux nouveaux groupages, donnant successivement accès à une classe d'équivalence maximale puis à une structure plus forte. En particulier, les liens entre les idéaux du demi-treillis engendré et les familles classiques d'automates cellulaires semblent prometteurs. Quelles améliorations espérer par l'introduction de nouvelles généralisations du groupage ?

Les travaux présentés dans cette thèse proposent d'aborder la problématique des interactions entre comportement local et global selon un angle structurel, à la fois en décrivant la structure interne des automates cellulaires, observés du point de vue de leur comportement global, mais aussi en structurant l'ensemble des automates cellulaires selon des critères pertinents dans ce cadre. Ainsi, différents raffinements du groupage sont introduits, structurant l'ensemble des automates cellulaires en le quotientant successivement par des transformations locales qui correspondent aux transferts simples d'informations locales vers le comportement global : l'utilisation de blocs de cellules dans le temps et dans l'espace, le produit cartésien d'automates cellulaires. Cette nouvelle structure unifie de plus un certain nombre de notions représentées par des familles classiques d'automates cellulaires. Enfin les interactions entre comportement local et global sont abordées par l'étude des automates cellulaires les plus complexes : les automates cellulaires intrinsèquement universels. En contraignant ces objets et en cherchant à minimiser certains paramètres, de nouvelles méthodes de transfert d'information du niveau local au niveau global sont mises à jour.

Cette approche structurelle mêlant des aspects algébriques, géométriques et algorithmiques doit être poursuivie en essayant de renforcer les interactions entre ces différents points de vue. Le développement de nouveaux outils algorithmiques de construction d'automates cellulaires complexes et la classification d'autres familles remarquables d'automates cellulaires devraient donner des pistes pour aller dans cette direction. Enfin, l'élargissement du champ d'étude et la confrontation des groupages à d'autres approches intéressantes des automates cellulaires, par exemple topologiques, permettront d'établir à plus long terme la pertinence et les limites de ces outils et de les améliorer.

CHAPITRE

1 AUTOMATES CELLULAIRES

On introduit ici les automates cellulaires, la problématique de classification et différentes façons de l'aborder. Après une brève présentation des définitions formelles traditionnelles des automates cellulaires, on discute leur représentation syntaxique, leurs propriétés dynamiques ainsi qu'une caractérisation originale par clôture. Puis, on présente diverses approches utilisées pour classifier les automates cellulaires, ainsi qu'un point de vue motivé par la calculabilité et la complexité structurelle.

Sommaire

1.1 Automates cellulaires . . . 19
1.1.1 L'objet et sa définition . . . 20
1.1.2 Discussion et propriétés . . . 20
Forme syntaxique . . . 21
Évolution . . . 24
1.1.3 Une caractérisation par clôture . . . 30
Cas général . . . 30
Cas des automates cellulaires réversibles . . . 31
1.2 Classifications . . . 33
1.2.1 Genèse . . . 33
L'approche expérimentale . . . 33
L'approche topologique . . . 36
L'approche algébrique . . . 36
1.2.2 Classification, calculabilité et complexité . . . 45
Indécidabilité de l'apparition d'un motif . . . 45
Complexité de vérification d'un triangle de calcul . . . 46
Limites de ces approches . . . 46

1.1 Automates cellulaires

On attribue généralement les premiers travaux sur les automates cellulaires à S. Ulam et J. von Neumann d'une part et à K. Zuse d'autre part. Dans les deux cas, il s'agissait de modéliser des phénomènes – l'auto-reproduction en biologie pour J. von Neumann [80], le monde physique pour K. Zuse [83] – par un réseau régulier d'entités uniformes simples interagissant localement de manière synchrone et déterministe. L'automate cellulaire décrit par J. von Neumann illustre bien une des problématiques de l'étude des systèmes complexes : le système est décrit localement, syntaxiquement, de manière synthétique et facilement compréhensible et possède un comportement global complexe (il modélise une certaine forme d'auto-reproduction) difficile à saisir. Les liens entre propriétés locales et globales sont au coeur de cette problématique.

1.1.1 L'objet et sa définition

Un automate cellulaire est défini par la donnée : d'un réseau régulier, d'un ensemble fini d'états utilisés pour colorier les sommets de ce réseau – les cellules – ainsi que d'une règle décrivant comment une cellule change d'état en fonction des états de ses voisines. Afin de passer d'une configuration – un coloriage des cellules par les états – à la suivante, la règle, qui est locale et déterministe, est appliquée uniformément, de manière synchrone, à toutes les cellules.

Traditionnellement, on considère les cellules comme étant les sommets d'une grille régulière et on définit le réseau d'interaction entre cellules par un ensemble fini de vecteurs donnant la position relative des voisins dans ce système de coordonnées.

Automate cellulaire. Un *automate cellulaire* $\mathcal{A}$ est un quadruplet $\left(\mathbb{Z}^d, Q, V, \delta\right)$ où l'entier d est la *dimension*, l'ensemble fini Q est l'*ensemble d'états*, la partie finie ordonnée V de $\mathbb{Z}^d$ est le *voisinage* et l'application δ de $Q^{|V|}$ dans Q est la *règle locale de transition* de l'automate cellulaire $\mathcal{A}$. Étant donné un automate cellulaire $\mathcal{A}$, on désignera par la suite par $Q_{\mathcal{A}}$, $V_{\mathcal{A}}$ et $\delta_{\mathcal{A}}$ respectivement l'ensemble d'états, le voisinage et la règle locale de transition de $\mathcal{A}$.

Évolution. Une *configuration* C d'un automate cellulaire $\mathcal{A}$ de dimension d est une application de $\mathbb{Z}^d$ dans $Q_{\mathcal{A}}$. On notera C_p l'état de la cellule de position p de C et $Q_{\mathcal{A}}^{\mathbb{Z}^d}$ l'ensemble des configurations de $\mathcal{A}$. La *règle globale de transition* G d'un automate cellulaire $\mathcal{A}$ est l'application de $Q_{\mathcal{A}}^{\mathbb{Z}^d}$ dans lui-même, qui à une configuration C de $\mathcal{A}$ associe la configuration $G(C)$ définie en toute position de la grille $p \in \mathbb{Z}^d$ par

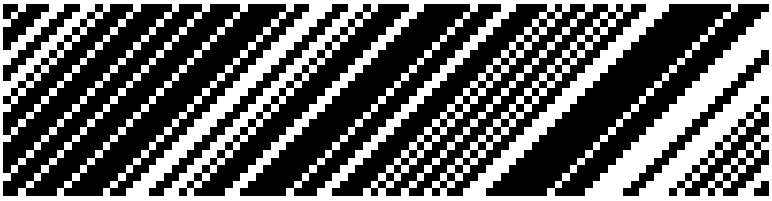
$$G(C)_p = \delta(C_{p+v_1}, \ldots, C_{p+v_\nu})$$

où $\{v_1, \ldots, v_\nu\}$ est le voisinage de $\mathcal{A}$. L'*évolution* Δ d'un automate cellulaire $\mathcal{A}$ partant de la configuration $C \in Q_{\mathcal{A}}^{\mathbb{Z}^d}$ est la suite $\left(\Delta^{(t)}\right)_{t\in\mathbb{N}}$ de configurations définie par $\Delta^{(0)} = C$ et, pour tout temps t non nul, $\Delta^{(t)} = G_{\mathcal{A}}\left(\Delta^{(t-1)}\right)$. On appelle *diagramme espace-temps* d'un automate cellulaire la représentation graphique d'une évolution obtenue en empilant de bas en haut les configurations obtenues au cours du temps.

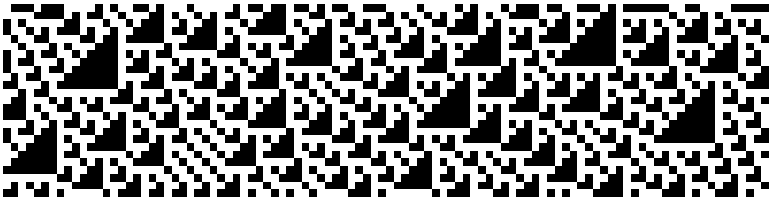
Exemple. La figure 1.1 présente trois automates cellulaires de dimension 1 ainsi que pour chacun d'entre eux un diagramme espace-temps partiel.

1.1.2 Discussion et propriétés

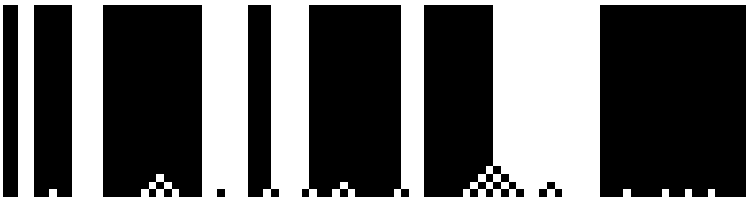
L'objet principal d'étude de cette thèse est l'ensemble des évolutions des automates cellulaires. Aussi, l'objet qui nous intéresse réellement n'est pas l'automate cellulaire mais le système dynamique constitué de l'ensemble des configurations et de la règle globale d'évolution. Nous discutons donc les conditions syntaxiques nécessaires et suffisantes pour que deux automates cellulaires aient un même ensemble d'évolutions. Enfin nous présentons quelques variantes et propriétés rattachées à ces définitions.



$\sigma = (\mathbb{Z}, \{\blacksquare, \square\}, \{-1\}, q \mapsto q)$



$\Sigma_2 = (\mathbb{Z}, \{\blacksquare, \square\}, [\![-1, 0]\!], (q, q') \mapsto q \oplus q')$,
où $(\{\blacksquare, \square\}, \oplus)$ est isomorphe à $(\mathbb{Z}_2, +)$



$(\mathbb{Z}, \{\blacksquare, \square\}, [\![-1, 1]\!], \mathsf{maj})$,
où maj retourne la valeur majoritaire parmi trois

FIG. 1.1 – Exemples de diagrammes espace-temps partiels

Forme syntaxique

La définition choisie pour les automates cellulaires fait certains choix de représentation. Nous discutons leurs variantes et leurs implications.

Voisinage. Le voisinage est défini comme un ensemble ordonné de vecteurs de déplacement dans la grille. Le choix d'un ordre sur les vecteurs de voisinage est un outil pour définir l'application de la règle locale de transition mais n'a pas d'incidence sur les évolutions de l'automate cellulaire.

► En effet, soit $\mathcal{A}$ un automate cellulaire de dimension d et de voisinage $\{v_1, \ldots, v_\nu\}$. Soit τ une permutation de $[\![1, \nu]\!]$. L'automate cellulaire $\mathcal{A}_\tau$ défini par $\left(\mathbb{Z}^d, Q_\mathcal{A}, \left\{v_{\tau(1)}, \ldots, v_{\tau(\nu)}\right\}, \delta_\tau\right)$, avec pour règle locale de transition $\delta_\tau : (x_1, \ldots, x_\nu) \mapsto \delta_\mathcal{A}(x_{\tau^{-1}(1)}, \ldots, x_{\tau^{-1}(\nu)})$, a même règle globale de transition que $\mathcal{A}$ (*i.e.* $G_\mathcal{A} = G_{\mathcal{A}_\tau}$).

Implicitement, on travaillera toujours à réordonnement des voisins près et on désignera parfois par voisinage l'ensemble des vecteurs de voisinage, indépendamment de l'ordre.

Le voisinage d'un automate cellulaire peut être augmenté sans changer la règle globale de transition. Il suffit d'ignorer les dépendances ajoutées.

► Formellement, soit $\mathcal{A}$ un automate cellulaire de dimension d et de voisinage $\{v_1, \ldots, v_\nu\}$. Soit v un vecteur de voisinage tel que v n'appartienne pas au voisinage de $\mathcal{A}$ (*i.e.* $v \notin V_\mathcal{A}$). L'automate cellulaire $\mathcal{A}_v$ défini par $\left(\mathbb{Z}^d, Q_\mathcal{A}, \{v_1, \ldots, v_\nu, v\}, \delta_v\right)$, avec pour règle locale de tran-

sition $\delta_v : (x_1, \ldots, x_{\nu+1}) \mapsto \delta_{\mathcal{A}}(x_1, \ldots, x_\nu)$, a même règle globale de transition que $\mathcal{A}$ (*i.e.* $G_{\mathcal{A}} = G_{\mathcal{A}_v}$).

Inversement, on peut supprimer les vecteurs de voisinages inutiles. En fait, étant donnée une règle globale de transition d'automate cellulaire G, il existe – à réordonnement des vecteurs de voisinage près – un unique automate cellulaire $\mathcal{A}$ dont le voisinage est minimal au sens de l'inclusion et dont la règle globale de transition est G.

▶ En effet, soit $\mathcal{B}$ un automate cellulaire de dimension d dont la règle globale de transition est G et soit $\{v_1, \ldots, v_\nu\}$ son voisinage. Un vecteur de voisinage v_i est utile s'il existe un $(\nu-1)$-uple d'états $(x_1, \ldots, x_{\nu-1})$ de $\mathcal{B}$ et deux états x et y de $\mathcal{B}$ tels que

$$\delta(x_1, \ldots, x_{i-1}, x, x_i, \ldots, x_{\nu-1}) \neq \delta(x_1, \ldots, x_{i-1}, y, x_i, \ldots, x_{\nu-1}) \quad .$$

L'ensemble des vecteurs de voisinage utiles $\{v_{i_1}, \ldots, v_{i_k}\}$ est le voisinage minimal de G. Soit q un état de $\mathcal{B}$. L'automate cellulaire $\mathcal{A}$ est alors défini par

$$\left(\mathbb{Z}^d, Q_{\mathcal{B}}, \{v_{i_1}, \ldots, v_{i_k}\}, (x_1, \ldots, x_k) \mapsto \delta_{\mathcal{B}}(y_1, \ldots, y_\nu)\right)$$

où y_j est égal à x_l si v_j est un vecteur utile v_{i_l} et à q sinon.

Implicitement, on associera toujours à une règle globale de transition l'automate cellulaire dont le voisinage est minimal. Une notion très pratique est celle de rayon de voisinage. On appelle rayon de voisinage d'un automate cellulaire $\mathcal{A}$, noté $r_{\mathcal{A}}$, le maximum des normes infinies de ses vecteurs de voisinages (*i.e.* $r_{\mathcal{A}} = \max\{\|v\|_\infty\}_{v \in V_{\mathcal{A}}}$). Par extension, on appelle rayon de voisinage minimal le rayon de voisinage de l'automate cellulaire réalisant le voisinage minimal.

Pour clore ce paragraphe consacré aux voisinages, la table 1.1 présente quelques voisinages classiques dont certains sont représentés sur la figure 1.2. À partir de la dimension 2, la notion de voisinage unidirectionnel est ambiguë, nous choisissons ici un voisinage inspiré par le voisinage de Moore.

notation	*nom*	*voisinage*
V^d_{OCA}	voisinage unidirectionnel	$\{-1, 0\}^d$
V^d_{Moore}	voisinage de Moore	$\{v \mid \|v\|_\infty \leqslant 1\}$
V^d_{vN}	voisinage de von Neumann	$\{v \mid \|v\|_1 \leqslant 1\}$
$V^d_\infty(n)$	voisinage carré de rayon n	$\{v \mid \|v\|_\infty \leqslant n\}$

TAB. 1.1 – Quelques voisinages classiques

Ensemble d'états. Le choix explicite d'un ensemble d'états n'est pas essentiel du point de vue des évolutions, les états pouvant être renommés pour peu qu'on adapte la règle locale de transition. Implicitement, on travaillera toujours à un renommage des états près.

Formellement, un automate cellulaire $\mathcal{A}$ est isomorphe à un automate cellulaire $\mathcal{B}$, noté $\mathcal{A} \cong \mathcal{B}$, s'il existe une application bijective $\varphi : Q_{\mathcal{A}} \to Q_{\mathcal{B}}$ telle que, si $\overline{\varphi}$ désigne l'extension canonique de φ à $Q_{\mathcal{A}}^{\mathbb{Z}^d}$, on a $\overline{\varphi} \circ G_{\mathcal{A}} = G_{\mathcal{B}} \circ \overline{\varphi}$.

En relâchant légèrement les contraintes de la définition de cette relation d'équivalence, en remplaçant la bijectivité de φ par de l'injectivité, on obtient une relation de préordre

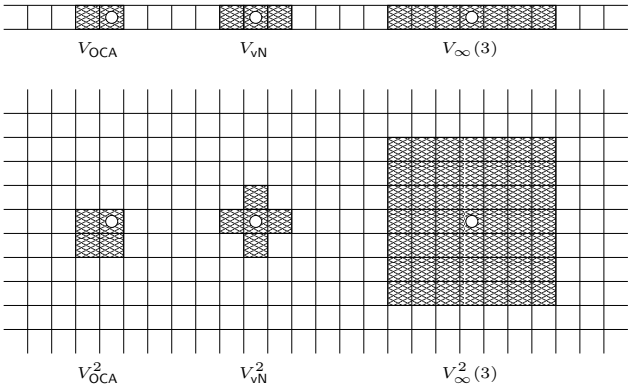


FIG. 1.2 – Quelques voisinages classiques

intéressante. Un automate cellulaire $\mathcal{A}$ est un sous-automate d'un automate cellulaire $\mathcal{B}$ si toute évolution de $\mathcal{A}$ est, à un renommage uniforme des états près, une évolution de $\mathcal{B}$.

Formellement, un automate cellulaire $\mathcal{A}$ est *sous-automate* d'un automate cellulaire $\mathcal{B}$, noté $\mathcal{A} \subseteq \mathcal{B}$, s'il existe une application injective $\varphi : Q_{\mathcal{A}} \to Q_{\mathcal{B}}$ telle que, si $\overline{\varphi}$ désigne l'extension canonique de φ à $Q_{\mathcal{A}}^{\mathbb{Z}^d}$, on a $\overline{\varphi} \circ G_{\mathcal{A}} = G_{\mathcal{B}} \circ \overline{\varphi}$. Autrement dit, le diagramme suivant commute :

$$\begin{array}{ccc} C & \xrightarrow{\overline{\varphi}} & \overline{\varphi}(C) \\ {\scriptstyle G_{\mathcal{A}}}\downarrow & & \downarrow{\scriptstyle G_{\mathcal{B}}} \\ G_{\mathcal{A}}(C) & \xrightarrow{\overline{\varphi}} & \overline{\varphi}(G_{\mathcal{A}}(C)) \end{array}$$

Règle locale de transition. En ajoutant des contraintes sur la règle locale de transition, on définit des familles classiques d'automates cellulaires.

Automates cellulaires partitionnés. Au lieu de transmettre tout son état à chacun de ses voisins, une cellule d'un automate cellulaire partitionné envoie un message différent à chaque voisin. L'ensemble d'états est un produit cartésien d'ensembles de messages. Chaque cellule commence par envoyer le i-ème message, la i-ème composante de son état, à la cellule dont elle est le i-ème voisin. Puis, une fois tous les messages expédiés, le nouvel état est calculé en fonction de tous les messages reçus. La règle locale de transition se décompose donc en une étape d'échange de messages et une étape de calcul.

Un *automate cellulaire partitionné* $\mathcal{A}$ est un quadruplet $\left\langle \mathbb{Z}^d, \prod_{i=1}^{\nu} Q_i, \{v_1, \ldots, v_\nu\}, \psi \right\rangle$ où les Q_i sont des ensembles finis de messages, les v_i sont des vecteurs de voisinage et ψ est une application de $\prod_{i=1}^{\nu} Q_i$ dans lui-même. L'automate cellulaire classique associé à $\mathcal{A}$ est $\left(\mathbb{Z}^d, \prod_{i=1}^{\nu} Q_i, \{v_1, \ldots, v_\nu\}, \delta\right)$ où δ est défini pour tout ν-uple d'états $\left(x^{(1)}, \ldots, x^{(\nu)}\right)$ par $\delta(x^{(1)}, \ldots, x^{(\nu)}) = \psi(x_1^{(1)}, \ldots, x_\nu^{(\nu)})$ où $x_i^{(i)}$ désigne le i-ème message du i-ème voisin. La règle globale de transition de $\mathcal{A}$ se décompose naturellement en une étape de transmission de messages suivie d'une étape de calcul

$$\overline{\psi} \circ (\sigma_{-v_1} \times \cdots \times \sigma_{-v_\nu}) \quad .$$

Automates cellulaires polynomiaux. En choisissant comme ensemble d'états une structure algébrique particulière et en prenant comme règle locale de transition une opération sur cette structure, on définit des automates cellulaires pour lesquels on espère pouvoir prédire plus de propriétés. Ainsi les automates cellulaires polynomiaux sont définis par une règle locale de transition polynomiale sur un anneau commutatif fini.

Un *automate cellulaire polynomial* $\mathcal{A}$ est un quadruplet $\left(\mathbb{Z}^d, (Q, +, \times), V, P\right)$ où V est un voisinage et P un polynôme en $|V|$ variables sur l'anneau commutatif fini $(Q, +, \times)$. L'automate cellulaire classique associé à $\mathcal{A}$ est $\left(\mathbb{Z}^d, Q, V, \delta\right)$ où δ est défini pour tout ν-uple d'états $(x_1, \ldots, x_\nu)$ par $\delta(x_1, \ldots, x_\nu) = P(x_1, \ldots, x_\nu)$.

Deux cas particuliers d'automates cellulaires polynomiaux seront étudiés dans cette thèse. Un *automate cellulaire linéaire* est un automate cellulaire polynomial dont le polynôme est de degré au plus un. Un *automate cellulaire bilinéaire* est un automate cellulaire polynomial dont le polynôme est composé uniquement de monômes de degré deux.

Évolution

L'ensemble des règles globales de transition des automates cellulaires est en bijection avec l'ensemble des automates cellulaires de voisinage minimal (quotienté par les permutations des vecteurs de voisinage). Nous discutons maintenant des variantes et propriétés des évolutions des automates cellulaires.

Configurations. En toute généralité, une configuration d'un automate cellulaire est infinie et non récursive donc pour nous inaccessible. Aussi est-il courant d'étudier la dynamique des automates cellulaires restreinte à des familles de configurations closes par application de la règle globale de transition et dont les diagrammes espace-temps sont raisonnablement représentables.

Configurations finies. Historiquement, les automates cellulaires ont souvent été étudiés sur les configurations finies[1]. Soit $\mathcal{A}$ un automate cellulaire de dimension d et q un état quiescent de $\mathcal{A}$ (*i.e.* vérifiant $\delta_{\mathcal{A}}(q, \ldots, q) = q$). Une *configuration q-finie* C de $\mathcal{A}$ est une configuration q-monochromatique sauf sur un support fini. Formellement, C vérifie

$$\exists r \in \mathbb{N}, \forall p \in \mathbb{Z}^d, \quad \|p\|_\infty \geqslant r \Rightarrow C_p = q \quad .$$

On notera $G^{\mathcal{F}}$ la restriction d'une règle globale de transition G aux configurations q-finies lorsque q est connu ou lorsque le choix de q est laissé libre.

Configurations périodiques. Dans cette thèse, en dimension d, on s'intéressera principalement aux configurations périodiques possédant d vecteurs de périodicité indépendants. Soit $\mathcal{A}$ un automate cellulaire de dimension d. Une *configuration périodique* C de $\mathcal{A}$, de période $\pi \in \mathbb{N}$, vérifie

$$\forall p \in \mathbb{Z}^d, \forall v \in [\![-1, 1]\!]^d, \quad C_{p+\pi v} = C_p \quad .$$

[1] Ce choix peut s'expliquer par une volonté de représentation cohérente du monde physique avec la présence d'un éther quiescent environnant. On peut aussi évoquer la facilité de conceptualisation, de représentation et de simulation numérique. Enfin, on peut relier ce choix aux configurations finies de certains modèles de calcul séquentiels, en particulier les machines de Turing.

On notera $G^{\mathcal{P}}$ la restriction d'une règle globale de transition G aux configurations périodiques. L'image par $G^{\mathcal{P}}$ d'une configuration périodique est une configuration périodique de période inférieure ou égale car l'automate cellulaire, de par sa définition uniforme, est incapable de briser la symétrie créée par la périodicité.

Configurations ultimement périodiques. Beaucoup moins répandues dans la littérature, les configurations ultimement périodiques représentent un compromis entre les deux types de configurations précédents. Soit $\mathcal{A}$ un automate cellulaire de dimension d. Une *configuration ultimement périodique* C de $\mathcal{A}$, de période π, est une configuration périodique sauf sur un support fini. Formellement, elle vérifie

$$\exists r \in \mathbb{N}, \forall p \in \mathbb{Z}^d, \forall v \in [\![-1,1]\!]^d, \quad (\|p\|_\infty \geqslant r \wedge \|p+\pi v\|_\infty \geqslant r) \Rightarrow C_p = C_{p+\pi v} \quad .$$

On notera $G^{\mathcal{U}}$ la restriction d'une règle globale de transition G aux configurations ultimement périodiques. Pour les mêmes raisons que ci-dessus, l'image par $G^{\mathcal{U}}$ d'une configuration ultimement périodique est une configuration ultimement périodique de période inférieure ou égale.

Configurations univers. Cette dernière famille de configurations est différente des précédentes. Elle n'est pas stable par application d'une règle globale de transition mais ces configurations possèdent des propriétés combinatoires intéressantes : elles contiennent tous les motifs finis possibles. Soit $\mathcal{A}$ un automate cellulaire de dimension d. Une *configuration univers* C de $\mathcal{A}$ est une configuration vérifiant

$$\forall C' \in Q_{\mathcal{A}}^{\mathbb{Z}^d}, \forall r \in \mathbb{N}, \exists p \in \mathbb{Z}^d, \forall v \in [\![-r,r]\!]^d, \quad C_{p+v} = C'_v \quad .$$

En particulier, si on connaît l'ensemble d'états d'un automate cellulaire $\mathcal{A}$ et un couple $(C, G_{\mathcal{A}}(C))$ formé d'une configuration univers et de son image par $G_{\mathcal{A}}$, alors on peut déterminer $\mathcal{A}$.

▶ En effet, on sait que tout motif fini apparaît dans C et que $\mathcal{A}$ possède un rayon de voisinage minimal. Pour trouver ce dernier, il suffit de trouver le plus petit r vérifiant

$$\forall p, p' \in \mathbb{Z}^d, \quad (\forall v \in [\![-r,r]\!]^d,\ C_{p+v} = C_{p'+v}) \Rightarrow G_{\mathcal{A}}(C)_p = G_{\mathcal{A}}(C)_{p'} \quad .$$

Puisque l'image d'un motif de rayon r n'est pas influencée par les états à l'extérieur du motif, r est un rayon de voisinage acceptable pour $\mathcal{A}$. Il suffit ensuite de construire δ en regardant l'image de chaque motif. L'universalité de C assure que tous les motifs sont présents.

Règle globale de transition. Nous explorons maintenant quelques propriétés classiques des règles globales de transition qui nous seront utiles par la suite.

Caractérisation topologique. Un résultat fondamental est la caractérisation topologique des automates cellulaires introduite par G. Hedlund [19] et redécouverte indépendamment par D. Richardson [24].

On munit $Q^{\mathbb{Z}^d}$ de la topologie produit, sur $\mathbb{Z}^d$, de la topologie triviale sur Q. On obtient ainsi un espace métrique compact. La translation de vecteur $v \in \mathbb{Z}^d$ sur $Q^{\mathbb{Z}^d}$ est l'application σ_v, bijective et continue, de $Q^{\mathbb{Z}^d}$ dans lui-même qui à une configuration C associe la configuration $\sigma_v(C)$ vérifiant, en toute position $p \in \mathbb{Z}^d$, l'égalité $\sigma_v(C)_{p+v} = C_p$.

Théorème 1 (G. Hedlund [19]). *Soit Q un ensemble fini. Une application de $Q^{\mathbb{Z}^d}$ dans lui-même commutant avec les translations est continue si et seulement si elle est la règle globale de transition d'un automate cellulaire.*

► Pour une démonstration rapide et élégante, le lecteur est renvoyé à la thèse de B. Durand [7]. L'idée de la preuve est de remarquer que les ensembles de configurations de la forme

$$\mathcal{O}_{p,r}(C) = \left\{ C' \in Q^{\mathbb{Z}^d} \mid \forall v \in \mathbb{Z}^d, \quad \|v\|_\infty \leqslant r \Rightarrow C'_{p+v} = C_{p+v} \right\}$$

forment une base d'ouverts pour $Q^{\mathbb{Z}^d}$.

Ce résultat permet de s'affranchir un peu plus de la définition syntaxique des automates cellulaires. En particulier, il permet de vérifier rapidement que certaines compositions de fonctions correspondent à des automates cellulaires. D'autre part, l'espace étant compact, on en déduit que si une règle globale de transition d'un automate cellulaire est bijective, alors cet automate cellulaire est réversible : il existe un automate cellulaire dont la règle globale de transition est l'inverse de la première.

Pour simplifier l'écriture, nous nous permettons à partir de maintenant de confondre parfois l'automate cellulaire et sa règle globale de transition. Par exemple, on parlera d'automate cellulaire injectif pour désigner un automate cellulaire dont la règle globale de transition est injective.

Propriétés immédiates. Par propriétés immédiates, il faut entendre les propriétés observables en ne regardant qu'un pas de calcul (*i.e.* en étudiant l'image de G). Ces propriétés ont été étudiées en détail dans la littérature.

La surjectivité des automates cellulaires a été reliée à l'injectivité sur les configurations finies par E. F. Moore et J. Myhill par un théorème connu sous le nom de théorème des jardins d'Eden. Dans sa formulation originelle, il concerne uniquement les automates cellulaires possédant un état quiescent. On appelle jardin d'Eden une configuration qui n'est pas accessible par la règle globale de transition : on peut la quitter mais on ne peut y accéder. Cette notion de jardin d'Eden amène naturellement à l'idée d'ensemble limite que nous étudierons au paragraphe suivant.

Théorème 2 (E. F. Moore [16] et J. Myhill [17]). *Un automate cellulaire est surjectif si et seulement si il est injectif sur les configurations finies.*

On notera que ce résultat admet une variante faible pour les automates cellulaires sans état quiescent. En effet, pour tout automate cellulaire de règle globale de transition G il existe une puissance G^k de cette règle globale qui possède un état quiescent q.

► Pour s'en convaincre, il suffit de considérer l'action de G sur les configurations monochromatiques. Pour tout état q, notons $\overline{q}$ la configuration q-monochromatique. Un automate cellulaire étant défini uniformément, il ne peut rompre la symétrie d'une configuration monochromatique. Pour tout état q, il existe un état q' tel que $G(\overline{q}) = \overline{q'}$. L'ensemble des états étant fini, il existe au moins un cycle de configurations monochromatiques. Soit k la longueur d'un tel cycle et $\overline{q}$ une configuration du cycle. Alors, $G^k(\overline{q}) = \overline{q}$ donc q est quiescent pour G^k.

Du théorème précédent on déduit que G^k est surjective si et seulement si elle est injective sur les configurations finies. Or, G est surjective (respectivement injective) si et seulement si G^k l'est. On en déduit que G est surjective si et seulement si G^k est injective sur les configurations finies.

Du théorème précédent on conclut que les automates cellulaires injectifs sont bijectifs. En raison de la remarque précédente, ce résultat est valide pour tous les automates cellulaires, avec ou sans état quiescent.

Corollaire 1. *La règle globale de transition d'un automate cellulaire est injective si et seulement si elle est bijective.*

Ces propriétés immédiates fournissent des exemples simples de propriétés globales des automates cellulaires qui ne sont pas locales. En effet, si S. Amoroso et Y. N. Patt [23] ont montré que ces propriétés étaient décidables en dimension 1, J. Kari [52] a montré qu'elles étaient indécidables dès la dimension deux. Ce résultat montre aussi que le comportement des automates cellulaires est plus compliqué en dimension deux qu'en dimension un.

Théorème 3 (S. Amoroso et Y. N. Patt [23]). *L'injectivité et la surjectivité des automates cellulaires de dimension un sont décidables.*

La démonstration de ce théorème repose sur l'étude de transducteurs réalisant une transition de l'automate cellulaire. Par un argument de combinatoire des mots, on ramène la décidabilité de la propriété à l'étude des images d'un ensemble de mots finis. K. Sutner [43] a proposé un algorithme en temps quadratique pour résoudre ce problème. Cette construction ne se généralise pas à la dimension deux.

Théorème 4 (J. Kari [52]). *L'injectivité et la surjectivité des automates cellulaires de dimension deux sont indécidables.*

La démonstration de ce théorème procède par réduction au problème de pavabilité du plan par un jeu de tuiles de Wang et aux pavages apériodiques introduits par R. Robinson [21]. Une conséquence de ce résultat est qu'il n'existe pas de méthode efficace pour énumérer ces automates cellulaires. Il n'existe pas de fonction récursive reliant le rayon minimal de voisinage d'un automate cellulaire réversible à celui de son inverse.

Dans ce contexte, les automates cellulaires partitionnés ont un meilleur comportement : un automate cellulaire partitionné $\left\langle \mathbb{Z}^d, \prod_{i=1}^{\nu} Q_i, \{v_1, \ldots, v_\nu\}, \psi \right\rangle$ est bijectif si et seulement si ψ est bijective.

► En effet, la règle globale de transition G de cet automate partitionné peut se décomposer en $\overline{\psi} \circ \left(\sigma_{v_1}^{-1} \times \sigma_{v_2}^{-1} \times \cdots \times \sigma_{v_\nu}^{-1}\right)$ où $\overline{\psi}$ est l'extension canonique de ψ à $\left(\prod_{i=1}^{\nu} Q_i\right)^{\mathbb{Z}^d}$ et où les $\sigma_{v_i}^{-1}$ ont pour ensemble d'états Q_i. Un produit de translations étant bijectif et continu, la règle globale de transition est bijective si et seulement si $\overline{\psi}$ l'est.

Ce résultat peut sembler contredire l'indécidabilité de la bijectivité à partir de la dimension 2 mais il n'en est rien. En effet, s'il est possible de simuler simplement tout automate cellulaire par un automate cellulaire partitionné et tout automate cellulaire bijectif par un automate cellulaire partitionné bijectif, il n'existe pas de transformation récursive qui à un automate cellulaire associe un automate cellulaire partitionné tel que le premier soit bijectif si et seulement si le second l'est.

La figure 1.3, extraite de la thèse de B. Durand [7], résume les implications connues entre les propriétés immédiates sur les différentes familles de configurations.

Propriétés asymptotiques. À l'opposé des propriétés immédiates, on trouve les propriétés asymptotiques qui concernent le comportement à très long terme des automates cellulaires. On s'intéresse aux configurations qui peuvent apparaître à n'importe quel instant dans un comportement de l'automate cellulaire.

$$
\begin{tikzcd}
G \text{ bijective} \arrow[d, shift right] & \arrow[r, Leftrightarrow] & G \text{ réversible} \arrow[r, Leftrightarrow] & G \text{ injective} \arrow[d, shift right] \\
G^{\mathcal{F}} \text{ bijective} \arrow[u, shift right, Rightarrow, not=] \arrow[d, Leftrightarrow] & \arrow[r, Rightarrow, not=] \arrow[l, Rightarrow, not=] \square & G^{\mathcal{P}} \text{ injective} \arrow[r, Leftrightarrow] & G^{\mathcal{P}} \text{ bijective} \arrow[u, shift right, Rightarrow, not=, "\triangle"] \arrow[d, shift right] \\
G^{\mathcal{F}} \text{ surjective} \arrow[d, shift right] & & \arrow[r, Rightarrow, not=, "\square"] \arrow[l, Rightarrow, not=] & G^{\mathcal{P}} \text{ surjective} \arrow[u, shift right, Rightarrow, not=] \arrow[d, shift right] \\
G^{\mathcal{F}} \text{ injective} \arrow[u, shift right, Rightarrow, not=] & \arrow[r, Leftrightarrow] & \Omega = Q^{\mathbb{Z}^d} \arrow[r, Leftrightarrow] & G \text{ surjective} \arrow[u, shift right, Rightarrow, not=, "\square"]
\end{tikzcd}
$$

$\triangle$ signifie que la relation est vérifiée en dimension un, mais ne l'est pas pour les dimensions supérieures ; $\square$ signifie que la relation est vérifiée en dimension un, la question est ouverte en dimension deux.

FIG. 1.3 – Propriétés de la règle globale de transition et familles de configurations

L'*ensemble limite* $\Omega(\mathcal{A})$ d'un automate cellulaire $\mathcal{A}$ est l'ensemble des configurations qui peuvent apparaître en tout temps d'un comportement de $\mathcal{A}$. C'est le plus grand ensemble de configurations pour l'inclusion sur lequel $G_{\mathcal{A}}$ est surjective. Formellement,

$$\Omega(\mathcal{A}) = \bigcap_{t \in \mathbb{N}} \operatorname{Im}\left(G_{\mathcal{A}}^t\right) \quad .$$

Cet ensemble a été étudié par K. Čulik II, J. Pachl et S. Yu [38]. En particulier, il est toujours bien défini et est soit constitué d'un singleton, soit infini. Dans le cas où il s'agit d'un singleton, l'automate cellulaire est dit *nilpotent* et cette unique configuration asymptotique est nécessairement monochromatique. De plus, si un automate cellulaire $\mathcal{A}$ est nilpotent, il existe une borne T telle que toute configuration C de $\mathcal{A}$ atteint cette configuration asymptotique en au plus T transitions (*i.e.* $G^T(Q_{\mathcal{A}}^{\mathbb{Z}^d})$ est un singleton). K.Čulik II, J. Pachl et S. Yu [38] présentent une démonstration topologique de cette propriété dont l'argument principal est qu'une intersection non vide d'ensembles compacts non vides est non vide. Nous proposons ici une démonstration combinatoire originale qui remplace les arguments de compacité par des extractions sur des ensembles dénombrables de configurations. Cette démonstration présente une technique pour extraire de tout diagramme espace-temps une configuration de l'ensemble limite.

► En effet, soit Δ une évolution de $\mathcal{A}$ partant d'une configuration univers et soit $\bar{q}$ la configuration monochromatique asymptotique.

S'il existe un temps T tel que la configuration $\Delta^{(T)}$ est la configuration $\bar{q}$, alors T convient pour la borne cherchée. En effet, l'automate cellulaire $G_{\mathcal{A}}^T$ transformant une de ses configurations univers en la configuration monochromatique $\bar{q}$, sa règle locale de transition est constante, associant à toute suite d'états l'état q.

Supposons que pour tout temps t la configuration $\Delta^{(t)}$ soit distincte de $\bar{q}$. On montre par un simple argument d'extraction qu'alors $\mathcal{A}$ ne peut pas être nilpotent. Puisqu'aucune configuration $\Delta^{(t)}$ n'est q-monochromatique et puisque pour tout entier r il existe un nombre fini de motifs cubiques d'états de côté r (il y en a exactement $|Q|^{(2r+1)^d}$), alors pour tout entier r il existe un motif $M^{(r)} : [\![-r, r]\!]^d \to Q$ centré sur un état distinct de q (*i.e.* $M_0^{(r)} \neq q$) qui se répète en une infinité de temps dans Δ :

$$\exists (t_i, p_i) \in \left(\mathbb{N} \times \mathbb{Z}^d\right)^{\mathbb{N}}, \quad \forall i, j \in \mathbb{N}, t_i \neq t_j \quad \wedge \quad \forall v \in [\![-r, r]\!]^d, \Delta_{p_i+v}^{(t_i)} = M_v^{(r)} \quad .$$

De la suite des $\left(M^{(r)}\right)_{r\in\mathbb{N}}$, on extrait une configuration C de la manière suivante. On extrait successivement les états de C dans des boules cubiques centrées en zéro et de rayon croissant. Pour tout entier i, posons $r_{0,i} = i$. On extrait successivement de chaque suite $(r_{k,i})_{i\in\mathbb{N}}$ une suite $(r_{k+1,i})_{i\in\mathbb{N}}$ d'indices tels que les motifs $M^{(r_{k+1,i})}$ admettent un même motif $C^{(k)}$ de rayon k centré en 0. Formellement, $\forall i \in \mathbb{N}, \forall v \in [\![-k,k]\!]^d, \quad M_v^{(r_{k+1,i})} = C_v^{(k)}$. L'état de la cellule de position p de C est donnée par $C_p^{(\|p\|_\infty)}$. La configuration C obtenue n'est pas q-monochromatique car elle est distincte de q en 0. Or, par construction, la configuration C appartient à l'ensemble limite de $\mathcal{A}$. En effet, pour toute durée t, pour chaque $C^{(k)}$, on peut choisir dans Δ un motif antécédent qui l'engendre en t transitions (car les $M^{(r)}$ apparaissent aussi loin dans le temps qu'on le veut). En appliquant la technique d'extraction sur la famille des antécédents des $C^{(k)}$, on obtient une configuration qui engendre C en t transitions.

Les automates nilpotents semblent avoir un comportement très simple : la configuration devient homogène en un temps borné. Cependant, ce temps n'est pas bornable récursivement en la taille de l'automate cellulaire. En effet, K. Čulik II, J. Pachl et S. Yu [38] ont montré que cette propriété est indécidable en dimension deux. Ce résultat a été étendu à la dimension un par J. Kari [44] exhibant ainsi une propriété globale simple mais indécidable des automates cellulaires de dimension un.

Théorème 5 (K. Čulik II, J. Pachl et S. Yu [38]). *La nilpotence des automates cellulaires de dimension deux est indécidable.*

Théorème 6 (J. Kari [44]). *La nilpotence des automates cellulaires de dimension un est indécidable.*

La démonstration procède ici aussi par réduction à un problème de pavabilité. Généralisant ce résultat, J. Kari [53] a démontré un analogue du théorème de Rice pour les propriétés des ensembles limite des automates cellulaires : toute propriété non triviale concernant les ensembles limite est indécidable. Un résultat du même type a récemment été proposé par J. Cervelle [14].

La notion de convergence vers un point fixe peut être adaptée à d'autres familles de configurations comme les configurations périodiques. K. Sutner [42] a montré que la convergence de toutes les configurations périodiques vers un point fixe est indécidable.

Théorème 7 (K. Sutner [42]). *Savoir si toutes les configurations périodiques d'un automate cellulaire de dimension un évoluent vers un point fixe est indécidable.*

Ce résultat, obtenu par simulation de machines de Turing sur automate cellulaire, a été généralisé par J. Mazoyer et I. Rapaport [70] au cas où toutes les configurations périodiques atteignent un même point fixe en utilisant une réduction à un problème de pavabilité. Nous parlerons alors de nilpotence par analogie avec le cas général. Un automate cellulaire $\mathcal{A}$ est *nilpotent pour les configurations périodiques* s'il existe une configuration périodique C telle que, pour toute configuration périodique C', il existe un temps t tel que $G_{\mathcal{A}}^t(C') = C$. La configuration C est nécessairement monochromatique.

Théorème 8 (J. Mazoyer et I. Rapaport [70]). *La nilpotence des automates cellulaires de dimension un sur les configurations périodiques est indécidable.*

Dans le cas de la nilpotence, il est facile de montrer qu'un automate cellulaire est nilpotent : il suffit d'exhiber une borne T pour la transitoire (*i.e.* telle que G^T soit constante). Dans le cas de la nilpotence sur les configurations périodiques, il est facile de montrer qu'un automate cellulaire n'est pas nilpotent sur les configurations périodiques : il suffit d'exhiber un cycle non q-monochromatique (*i.e.* une configuration périodique C, distincte

de q en 0 et une durée T telle que $G^T(C) = C$). L'ensemble des automates cellulaires nilpotents est récursivement énumérable alors que l'ensemble des automates cellulaires nilpotents pour les configurations périodiques est co-récursivement énumérable.

1.1.3 Une caractérisation par clôture

Si la caractérisation topologique des automates cellulaires de G. Hedlund et D. Richardson permet de composer des automates cellulaires pour en fabriquer de nouveaux sans se soucier de passer par une règle locale de transition, il est encore nécessaire de définir les automates cellulaires de départ. Nous proposons ici une nouvelle caractérisation par clôture des automates cellulaires qui permet de s'affranchir complètement de la description locale. Nous en déduisons naturellement une caractérisation des automates cellulaires réversibles qui donne une présentation originale des automates cellulaires réversibles et simplifie leur construction.

Cas général

Nous montrons que tout automate cellulaire d'ensemble d'états de type $[\![1, n]\!]$ peut être obtenu par clôture d'une famille d'automates cellulaires aux comportements simples par les opérations de composition, produit cartésien et sous-automate.

Automate cellulaire autarcique. Un *automate cellulaire autarcique* $\mathcal{A}$ est un automate cellulaire de voisinage $\{0\}$. Chaque cellule évolue indépendamment des autres en utilisant une même règle locale de transition $\psi : Q_{\mathcal{A}} \to Q_{\mathcal{A}}$. On note $\overline{\psi}$ la règle globale de transition de cet automate cellulaire. Un tel automate cellulaire est ultimement périodique.

Translation élémentaire. Une *translation élémentaire* σ_v de dimension d est une translation d'un vecteur de base de $\mathbb{Z}^d$ ou de son opposé (*i.e.* $\|v\|_1 = 1$). L'ensemble des translations élémentaires de dimension d engendre toutes les translations de dimension d par composition.

Composition. La *composition* $\mathcal{A} \circ \mathcal{B}$ de deux automates cellulaires $\mathcal{A}$ et $\mathcal{B}$ de même ensemble d'états est l'automate cellulaire dont la règle globale de transition est la composition de leurs règles globales de transition (*i.e.* $G_{\mathcal{A}\circ\mathcal{B}} = G_{\mathcal{A}} \circ G_{\mathcal{B}}$). La composition de deux automates n'est définie que si ceux-ci ont même ensemble d'états.

Produit cartésien. Le *produit cartésien* $\mathcal{A} \times \mathcal{B}$ de deux automates cellulaires $\mathcal{A}$ et $\mathcal{B}$ est l'automate cellulaire dont la règle globale de transition est le produit cartésien de leurs règles globales de transition. Nous allons travailler uniquement avec des alphabets de type $[\![1, n]\!]$, aussi nous utiliserons implicitement une fonction de codage de $[\![1, m]\!] \times [\![1, n]\!]$ dans $[\![1, mn]\!]$ que nous noterons γ. Formellement, notre produit cartésien est défini par $G_{\mathcal{A}\times\mathcal{B}} = \overline{\gamma} \circ (G_{\mathcal{A}} \times G_{\mathcal{B}}) \circ \overline{\gamma^{-1}}$. Pour simplifier les notations, nous notons $\mathcal{A} \times \mathcal{B} \times \mathcal{C}$ le produit $\mathcal{A} \times (\mathcal{B} \times \mathcal{C})$ et nous étendons γ à tout produit cartésien d'un nombre fini d'ensembles de $\prod_{i=1}^{k} [\![1, n_i]\!]$ dans $[\![1, \prod_{i=1}^{k} n_i]\!]$.

Sous-automate. Le *sous-automate* à k états $\lfloor \mathcal{A} \rfloor_k$ d'un automate cellulaire $\mathcal{A}$ de dimension d et d'ensemble d'états $[\![1,n]\!]$ est l'automate cellulaire dont la règle globale de transition est la restriction de $G_{\mathcal{A}}$ à l'ensemble $[\![1,k]\!]^{\mathbb{Z}^d}$ (*i.e.* $G_{\lfloor \mathcal{A} \rfloor_k} = G_{\mathcal{A}}$). À une permutation des états près, tout sous-automate de $\mathcal{A}$ peut être obtenu ainsi. Le sous-automate n'est défini que lorsque la restriction à $[\![1,k]\!]$ de la règle de transition est bien à image dans $[\![1,k]\!]$.

Théorème 9. *Soit n un entier. Une application de $[\![1,n]\!]^{\mathbb{Z}^d}$ dans lui-même est la règle globale de transition d'un automate cellulaire si et seulement si elle appartient à la clôture des règles globales de transition des automates cellulaires autarciques et des translations élémentaires par les opérations de composition, produit cartésien et sous-automate.*

Démonstration. Les automates cellulaires autarciques et les translations élémentaires sont des automates cellulaires. De plus, les opérations de composition, produit cartésien et sous-automate appliquées à des automates cellulaires produisent des automates cellulaires. Aussi, tout élément de la clôture est bien un automate cellulaire.

Réciproquement, soit $\mathcal{A}$ un automate cellulaire de dimension d, d'ensemble d'états $[\![1,n]\!]$ et de voisinage $\{v_1,\ldots,v_\nu\}$. Considérons l'automate cellulaire partitionné

$$\mathcal{A}_p = \left\langle \mathbb{Z}^d, [\![1,n]\!]^\nu, \{v_1,\ldots,v_\nu\}, c_\nu \circ \delta_{\mathcal{A}} \right\rangle$$

où c_ν est l'application de $[\![1,n]\!]$ dans $[\![1,n]\!]^\nu$ définie pour tout état x par $c_\nu(x) = (x,\ldots,x)$. Par construction $\mathcal{A}$ est un sous-automate de $\mathcal{A}_p$ où c_ν est l'application injective de $Q_{\mathcal{A}}$ dans $Q_{\mathcal{A}_p}$ qui convient (*i.e.* $\overline{c_\nu} \circ G_{\mathcal{A}} = G_{\mathcal{A}_p} \circ \overline{c_\nu}$). Soit π une extension de $\gamma \circ c_\nu$ en une application bijective de $[\![1,n^\nu]\!]$ dans lui-même (*i.e.* vérifiant pour tout $x \in [\![1,n]\!]$ l'égalité $\pi(x) = \gamma(c_\nu(x))$). Soit G la règle globale de transition définie par $G = \overline{\gamma} \circ G_{\mathcal{A}_p} \circ \overline{\gamma^{-1}}$. On peut alors exprimer $G_{\mathcal{A}}$ de la manière suivante

$$G_{\mathcal{A}} = \left\lfloor \overline{\pi^{-1}} \circ G \circ \overline{\pi} \right\rfloor_n \quad .$$

Il suffit donc de montrer que G est exprimable dans la clôture. Comme $\mathcal{A}_p$ est un automate partitionné, sa règle globale de transition peut être décomposée en une étape d'envoi de messages suivie de l'application autarcique de la fonction de transition, ici $c_\nu \circ \delta_{\mathcal{A}}$. L'étape d'envoi des messages est un produit de translations. On sait que les translations sont toutes exprimables dans la clôture. Posons $\psi = \gamma \circ c_\nu \circ \delta_{\mathcal{A}} \circ \gamma^{-1}$. On peut alors exprimer la règle globale de transition de $\mathcal{A}$ de la façon suivante

$$G_{\mathcal{A}} = \left\lfloor \overline{\pi^{-1}} \circ \overline{\psi} \circ (\sigma_{-v_1} \times \cdots \times \sigma_{-v_\nu}) \circ \overline{\pi} \right\rfloor_n \quad .$$

Ce qui conclut la démonstration. ■

Cas des automates cellulaires réversibles

Une manière naïve d'énumérer les automates cellulaires réversibles consiste à énumérer toutes les paires d'automates cellulaires et à tester pour chaque paire $(\mathcal{A},\mathcal{B})$ si la composition $\mathcal{A} \circ \mathcal{B}$ est l'automate cellulaire identité (*i.e.* l'automate cellulaire $\mathcal{I}$ vérifiant pour toute configuration C l'égalité $G_{\mathcal{I}}(C) = C$), auquel cas l'automate cellulaire $\mathcal{A}$ est réversible et $\mathcal{B}$ est un inverse possible pour $\mathcal{A}$. Certes cette méthode est intrinsèquement inefficace, mais le résultat d'indécidabilité de J. Kari [39] assure qu'il n'existe aucune

méthode efficace. Cependant, nous proposons ici une nouvelle caractérisation des automates cellulaires réversibles qui nous semble plus satisfaisante pour la compréhension de la réversibilité.

Considérons la caractérisation du paragraphe précédent. Les trois opérations de composition, produit cartésien et sous-automate appliquées à des automates cellulaires réversibles produisent des automates cellulaires réversibles.

- En effet, pour les opérations de composition et de produit cartésien, c'est une conséquence évidente du théorème de G. Hedlund [19]. Pour l'opération de sous-automate, c'est une conséquence du corollaire 1. En effet, comme un automate cellulaire réversible est injectif, son sous-automate est nécessairement injectif et donc lui aussi réversible.

De ces considérations, on déduit qu'en conservant uniquement les translations élémentaires et les automates cellulaires autarciques réversibles comme objets de base, la clôture engendre un sous-ensemble des automates cellulaires réversibles. Nous montrons qu'en fait ils sont tous engendrés.

Théorème 10. *Soit n un entier. Une application de $[\![1,n]\!]^{\mathbb{Z}^d}$ dans lui-même est la règle globale de transition d'un automate cellulaire réversible si et seulement si elle appartient à la clôture des règles globales de transition des automates cellulaires autarciques réversibles et des translations élémentaires par les opérations de composition, produit cartésien et sous-automate.*

Démonstration. L'une des implications vient d'être discutée dans le texte. Considérons la seconde implication. soit $\mathcal{A}$ un automate cellulaire réversible de dimension d, d'ensemble d'états $[\![1,n]\!]$ et de voisinage $\{v_1,\dots,v_\nu\}$. Soit $\mathcal{B}$ l'automate cellulaire inverse de l'automate $\mathcal{A}$. Sans rien changer à la discussion, on peut supposer que $\{v_1,\dots,v_\nu\}$ est aussi le voisinage de $\mathcal{B}$ et que $v_1=0$ car dans le cas contraire il suffit de recommencer avec un voisinage plus grand pour $\mathcal{A}$, quitte à ajouter des vecteurs de voisinage inutiles. Considérons l'automate cellulaire partitionné

$$\mathcal{A}_p = \left\langle \mathbb{Z}^d, \left([\![1,n+1]\!]^2\right)^\nu, \{v_1,\dots,v_\nu\}, \psi \right\rangle$$

où ψ est défini partiellement de la manière suivante :

Transition. Pour tout ν-uple d'états $(x_1,\dots,x_\nu)$ de $[\![1,n]\!]$, posons $y=\delta_{\mathcal{A}}(x_1,\dots,x_\nu)$,

$$\psi\left((x_1,n+1),(x_2,n+1)\dots,(x_\nu,n+1)\right) = \left((x_1,y),(x_2,y)\dots,(x_\nu,y)\right) \quad .$$

Voisinage. Pour tout ν-uple d'états $(y_1,\dots,y_\nu)$ de $[\![1,n]\!]$, posons $x=\delta_{\mathcal{B}}(y_1,\dots,y_\nu)$,

$$\psi\left((x,y_1),(x,y_2),\dots,(x,y_\nu)\right) = \left((n+1,y_1),(n+1,y_2)\dots,(n+1,y_\nu)\right) \quad .$$

Nettoyage. Pour tout état y de $[\![1,n]\!]$,

$$\psi\left((n+1,y),(n+1,y),\dots,(n+1,y)\right) = \left((y,n+1),(y,n+1)\dots,(y,n+1)\right) \quad .$$

L'application partielle ψ ainsi définie est injective, on la complète en une application bijective. L'automate cellulaire partitionné $\mathcal{A}_p$ est donc réversible. Soit $\mathcal{B}_p$ l'automate cellulaire partitionné réversible défini comme $\mathcal{A}_p$ mais avec pour voisinage $\{-v_1,\dots,-v_\nu\}$. Soit c_ν l'application injective de $[\![1,n]\!]$ dans $([\![1,n+1]\!]^2)^\nu$ définie pour tout état x par

$c_\nu(x) = ((x, n+1), \ldots, (x, n+1))$. Par construction, $\mathcal{A}$ est un sous-automate de l'automate produit $\mathcal{A}_p \circ \mathcal{B}_p \circ \mathcal{A}_p$ où c_ν est l'application injective qui convient. Soit π une extension de $\gamma \circ c_\nu$ en une application bijective de $[\![1, (n+1)^{2\nu}]\!]$ dans lui-même (*i.e.* vérifiant pour tout $x \in [\![1, n]\!]$ l'égalité $\pi(x) = \gamma(c_\nu(x))$). Soit G la règle globale de transition définie par $G = \overline{\gamma} \circ G_{\mathcal{A}_p} \circ G_{\mathcal{B}_p} \circ G_{\mathcal{A}_p} \circ \overline{\gamma^{-1}}$. On peut alors exprimer $G_{\mathcal{A}}$ de la manière suivante

$$G_{\mathcal{A}} = \left\lfloor \overline{\pi^{-1}} \circ G \circ \overline{\pi} \right\rfloor_n \quad .$$

Il suffit donc de montrer que G est exprimable dans la clôture. Pour cela, on va montrer que $G' = \overline{\gamma} \circ G_{\mathcal{A}_p} \circ \overline{\gamma^{-1}}$ est exprimable, car le cas de $G'' = \overline{\gamma} \circ G_{\mathcal{B}_p} \circ \overline{\gamma^{-1}}$ est symétrique et on a $G = G' \circ G'' \circ G'$. Comme $\mathcal{A}_p$ est un automate partitionné, sa règle globale de transition peut être décomposée en une étape d'envoi de messages suivie de l'application autarcique de la fonction de transition, ici ψ. L'étape d'envoi des messages est un produit de translations. On sait que les translations sont toutes exprimables dans la clôture. Posons $\psi' = \gamma \circ \psi \circ \gamma^{-1}$. La fonction $\overline{\psi'}$ est la règle globale de transition d'un automate cellulaire autarcique réversible car ψ est bijective. On peut alors exprimer la règle globale de transition de $\mathcal{A}$ de la façon suivante

$$\begin{aligned} G_{\mathcal{A}} = \Big\lfloor & \overline{\pi^{-1}} \circ \overline{\psi'} \circ (\sigma_{-v_1} \times \cdots \times \sigma_{-v_\nu}) \\ & \circ \overline{\psi'} \circ (\sigma_{\ v_1} \times \cdots \times \sigma_{\ v_\nu}) \\ & \circ \overline{\psi'} \circ (\sigma_{-v_1} \times \cdots \times \sigma_{-v_\nu}) \circ \overline{\pi} \Big\rfloor_n \quad . \end{aligned}$$

Ce qui conclut la démonstration. ■

Alors que dans la caractérisation du paragraphe précédent on peut prendre pour voisinage le voisinage minimal de l'automate cellulaire et ainsi borner la taille de la plus petite expression qui le décrit, ici le voisinage requis est inconnu de par l'indécidabilité de la réversibilité. Aussi, il est indécidable de savoir si une expression dans la première clôture est exprimable dans la seconde. En outre, lorsque cette opération est possible, elle peut faire croître la taille de l'expression dans des proportions non bornées.

1.2 Classifications

Une des motivations principales de l'étude mathématique des automates cellulaires est de comprendre et de maîtriser leurs évolutions afin, entre autre, de fournir des outils pertinents pour étudier les automates cellulaires qui apparaissent dans un cadre de modélisation. Une approche consiste à établir des classifications de ces objets qui tiennent compte de leurs propriétés, séparant les automates cellulaires simples des automates cellulaires complexes, pour une certaine idée de complexité.

1.2.1 Genèse

L'approche expérimentale

La première proposition de classification est certainement celle présentée en 1984 par S. Wolfram [29]. Cette classification résulte d'une approche expérimentale des automates cellulaires. Voici comment l'auteur présente cette classification dans un récent ouvrage :

« [. . .] Indeed, among all kinds of cellular automata, it seems that the patterns which arise can almost always be assigned quite easily to one of just four basic classes [. . .]

These classes are conveniently numbered in order of increasing complexity, and each one has certain immediate distinctive features.

In class 1, the behavior is very simple, and almost all initial conditions lead to exactly the same uniform final state.

In class 2, there are many different possible final states, but all of them consist just of a certain set of simple structures that either remain the same forever or repeat every few steps.

In class 3, the behavior is more complicated, and seems in many respects random, although triangles and other small-scale structures are essentially always at some level seen.

And finally [...] class 4 involves a mixture of order and randomness: localized structures are produced which on their own are fairly simple, but these structures move around and interact with each other in very complicated ways. [. . .] »

S. Wolfram [91, chapitre 6, pp. 231–235]

L'étude de S. Wolfram se place dans un cadre précis. Ses constatations sont fondées sur l'observation de diagrammes espace-temps (nécessairement partiels) d'automates cellulaires de dimension 1, à deux états et, en particulier, les 256 automates à voisinage de von Neumann, les automates cellulaires élémentaires, dont des diagrammes espace-temps sont présentés sur la figure 1.4. De ces observations et de nombreuses autres, S. Wolfram exhibe un partitionnement des automates cellulaires en quatre classes :

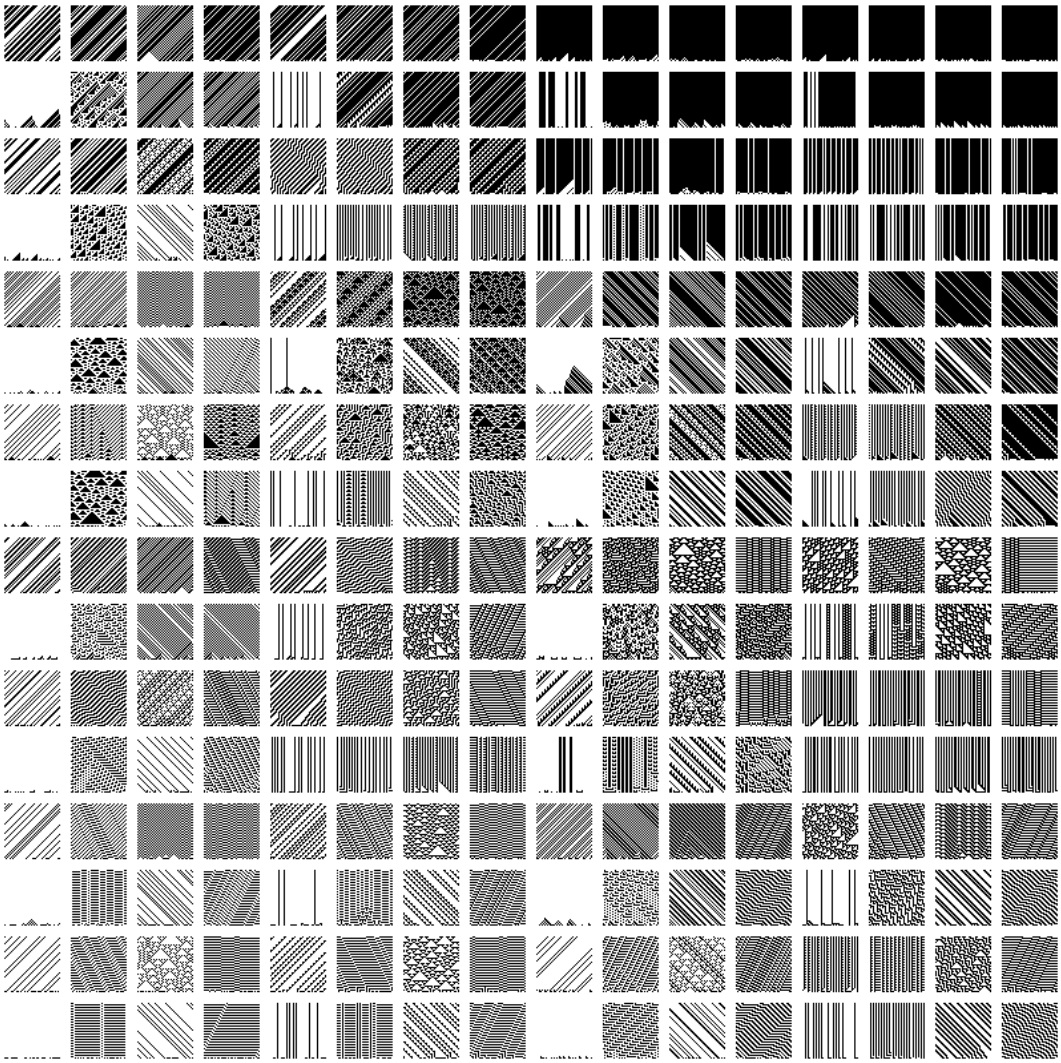


Fig. 1.4 – Diagrammes espace-temps des 256 automates cellulaire élémentaires

Classe 1 presque toutes les conditions initiales convergent en temps fini vers un même état uniforme ;

Classe 2 presque toutes les conditions initiales convergent en temps fini vers des états éventuellement distincts mais tous périodiques dans le temps ;

Classe 3 il n'y a pas de convergence simple visible, l'évolution semble être aléatoire mis à part une éventuelle micro-structuration ;

Classe 4 rapidement des structures simples et localisées se mettent en place, qui interagissent entre elles de manière compliquée.

Cette classification a l'avantage de distinguer deux notions de complexité distinctes : la classe 3 correspond aux automates cellulaires «chaotiques», ceux dont le comportement est dit complexe car difficilement prévisible et la classe 4 correspond aux automates cellulaires complexes, ceux dont le comportement est dit complexe car il en émerge des interactions très structurées qui semblent algorithmiquement complexes. Cependant, cette classification n'est pas très satisfaisante. Tout d'abord, ce sont les paires constituées d'un automate cellulaire et d'une configuration initiale qui sont classifiées et non les automates cellulaires eux-mêmes. D'autre part, le manque de formalisation de cette classification la rend inutilisable en pratique : S. Wolfram lui-même classe certains automates cellulaires dans plusieurs classes dans ses publications.

K. Čulik II et S. Yu [37] ont proposé une formalisation de cette classification et ont montré que l'appartenance aux classes est indécidable. Les classes sont définies ainsi :

Classe 1 toute configuration initiale finie converge vers la configuration uniformément quiescente en un temps fini ;

Classe 2 toute configuration initiale finie a une évolution ultimement périodique ;

Classe 3 pour toute paire de configurations finies, savoir si l'automate cellulaire atteint la seconde configuration à partir de la première en un temps fini est décidable ;

Classe 4 tous les autres automates cellulaires.

Bien formalisée, cette classification présente toutefois de sérieux inconvénients. On considère des configurations finies, ce qui distingue un état quiescent particulier, on classifie donc plutôt un couple constitué d'un automate cellulaire et d'un état quiescent. Les automates cellulaires sans état quiescent atteignent alors la classe 4 censée être de complexité maximale. D'autre part, si l'automate cellulaire considéré force une certaine monotonie sur les configurations finies (si la croissance de la taille du motif non quiescent d'une configuration finie est strictement croissante ou strictement décroissante), l'automate se retrouve au plus dans la classe 3 voire dans la classe 2. Enfin, des automates comme les translations se retrouvent dans la classe 3 car leur comportement est périodique à une translation près et donc non périodique au sens strict du terme.

D'autres propositions ont été faites pour formaliser la classification de S. Wolfram. Notre but n'étant pas ici d'être exhaustifs, nous nous contenterons de résumer les problèmes clés rencontrés dans ces approches. Un premier point concerne les classes 3 et 4, les automates cellulaires chaotiques et complexes : que veut-on mettre dans ces classes ? Un second point est de savoir ce qu'on veut réellement classifier : les automates cellulaires bien sûr mais sur quelle famille de configurations ? Un dernier point enfin concerne le problème de la classification des translations : on ne désire pas les trouver au-dessus de la

classe 2 ; de même pour un produit cartésien de translations car ces automates cellulaires engendrent des comportements très simples.

L'approche topologique

La classe 3 de la classification de S. Wolfram est considérée comme la classe des automates cellulaires chaotiques. Des exemples typiques de tels automates cellulaires sont présentés sur la figure 1.5. De nombreuses classifications ont été proposées pour mieux cerner ces automates cellulaires.

Une des premières propositions dans ce sens est celle de R. H. Gilman [35]. L'idée intéressante de cette classification est de reprendre les résultats obtenus en théorie des systèmes dynamiques et de les adapter au cas des automates cellulaires. Pour cela, l'ensemble des configurations est muni de la distance associée à la topologie de Cantor (topologie produit sur $\mathbb{Z}$ de la topologie triviale sur l'ensemble d'états), définie pour tout couple de configurations C, C', par $d(C, C') = 1/2^{\min\left\{\|p\|_\infty | C_p \neq C'_p\right\}}$. Grâce à cette distance, on peut exprimer les notions d'équicontinuité et d'expansivité[2]. R. H. Gilman définit, à l'aide de ces notions et de la théorie de la mesure, une classification en trois classes :

Classe 1 équicontinuité en certains points ;

Classe 2 équicontinuité presque partout mais pas classe 3 ;

Classe 3 expansivité presque partout.

On peut aussi noter l'introduction plus récente, par P. Kůrka [61], d'une classification purement topologique, sans mesure, en quatre classes :

Classe 1 équicontinuité ;

Classe 2 des points d'équicontinuité mais pas classe 1 ;

Classe 3 sensibilité aux conditions initiales ;

Classe 4 expansivité.

Le principal problème de ces classifications dans le cadre de la topologie de Cantor est la classification des translations dans les classes correspondant aux automates cellulaires les plus complexes. Pour résoudre ce problème, G. Cattaneo *et al.* [63] ont proposé des définitions plus fines des automates cellulaires chaotiques en conservant la topologie de Cantor. Indépendamment, E. Formenti [11] a défini une autre topologie, dite topologie de Besicovitch, dans laquelle les classifications précédentes sont beaucoup plus satisfaisantes. Dans le prolongement de ces travaux[3], B. Martin II [13] a étudié les avantages de réintroduire les outils de théorie de la mesure utilisés pas R. H. Gilman dans le cadre de cette nouvelle topologie.

L'approche algébrique

La classe 4 de la classification de S. Wolfram est considérée comme la classe des automates complexes dans les évolutions desquels apparaissent des structures simples et

[2]La définition formelle de ces notions dépassant le cadre de cette thèse, le lecteur intéressé est invité à se reporter à la thèse de E. Formenti [11].

[3]Une autre voie prometteuse est la nouvelle topologie reposant sur la complexité de Kolmogorov introduite par J. Cervelle [14].

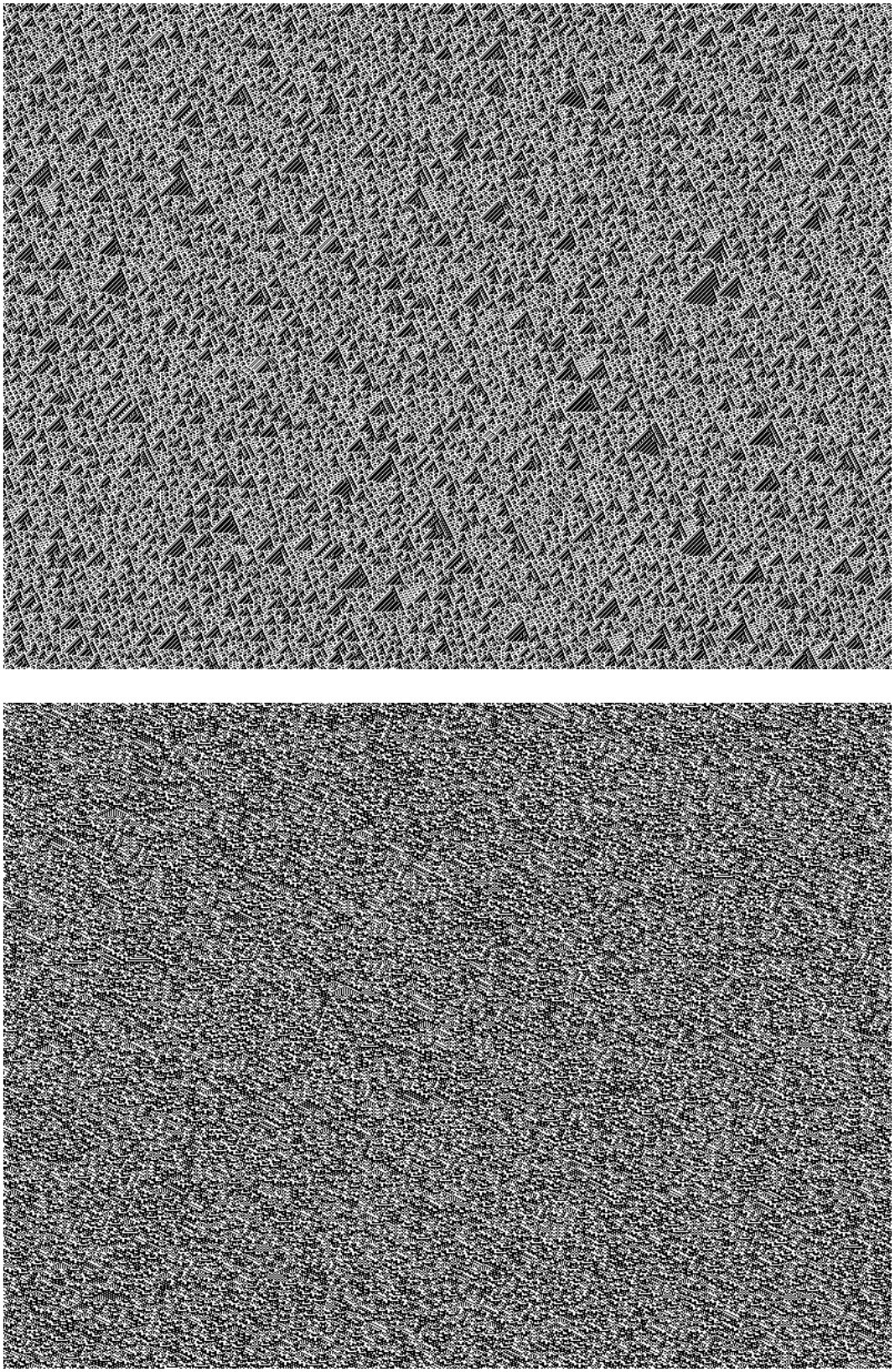


Fig. 1.5 – Diagrammes espace-temps d'automates cellulaires chaotiques (1/2)

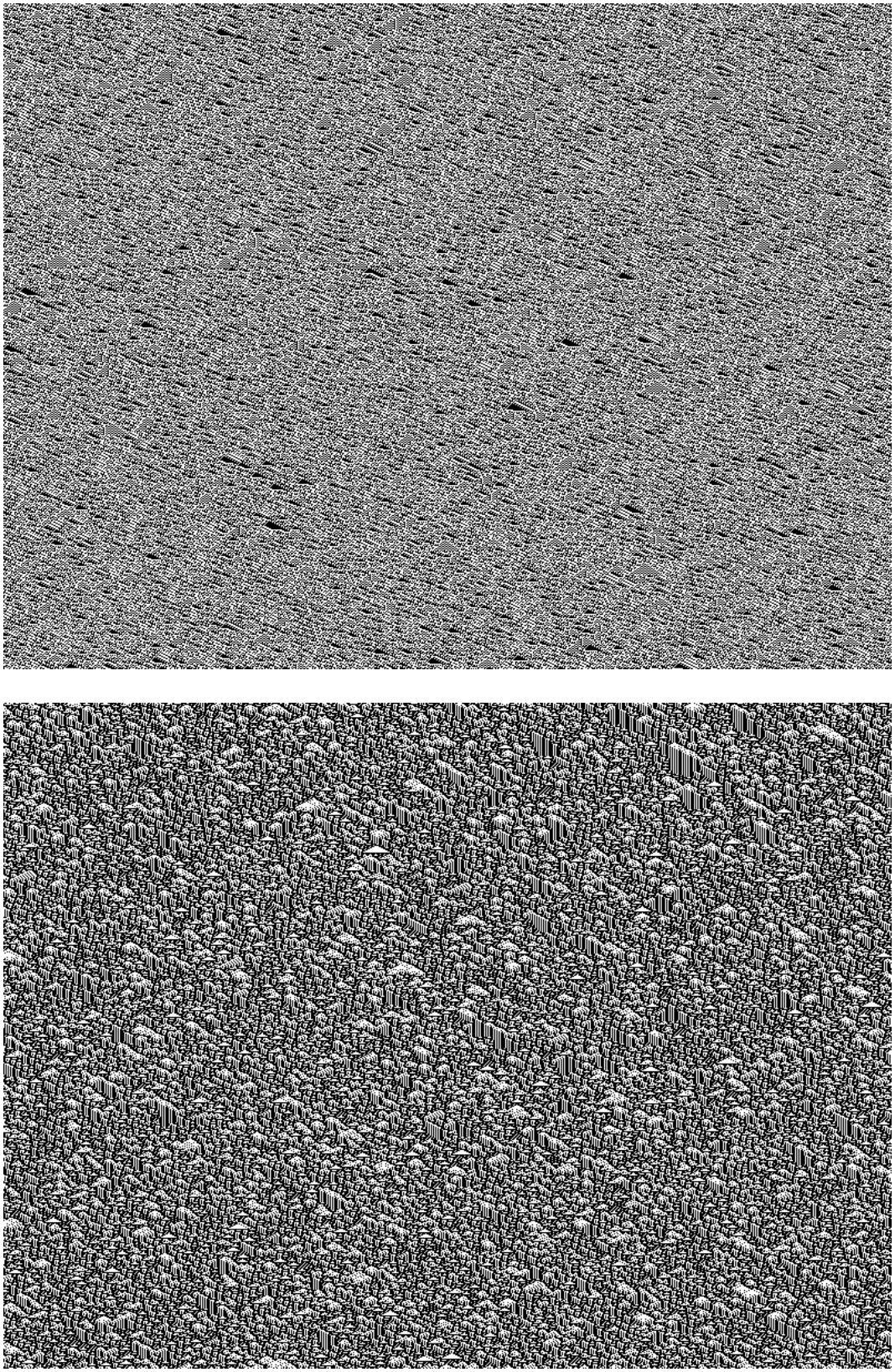


FIG. 1.5 – Diagrammes espace-temps d'automates cellulaires chaotiques (2/2)

localisées qui interagissent entre elles de manière compliquée. Des exemples typiques de tels automates cellulaires sont présentés sur la figure 1.6. À notre connaissance, le groupage d'I. Rapaport [12] est la seule tentative de classification qui se focalise sur ces automates cellulaires.

Au lieu d'opter pour une approche topologique, comme pour la classification des automates cellulaires selon leur chaoticité, l'approche est ici algébrique. On introduit une relation de préordre sur l'ensemble des automates cellulaires. Les classes de la classification sont alors les classes d'équivalence de la relation d'équivalence associée et la hiérarchie entre les classes est la relation d'ordre induite sur les classes.

Pour définir ce préordre, on considère la relation sous-automate qui est une façon algébrique naturelle de comparer deux automates cellulaires. Cependant, au lieu d'appliquer cette relation directement aux automates cellulaires à comparer, on permet certaines transformations algébriques préalables de ces automates cellulaires qui ont un sens pour les automates cellulaires complexes.

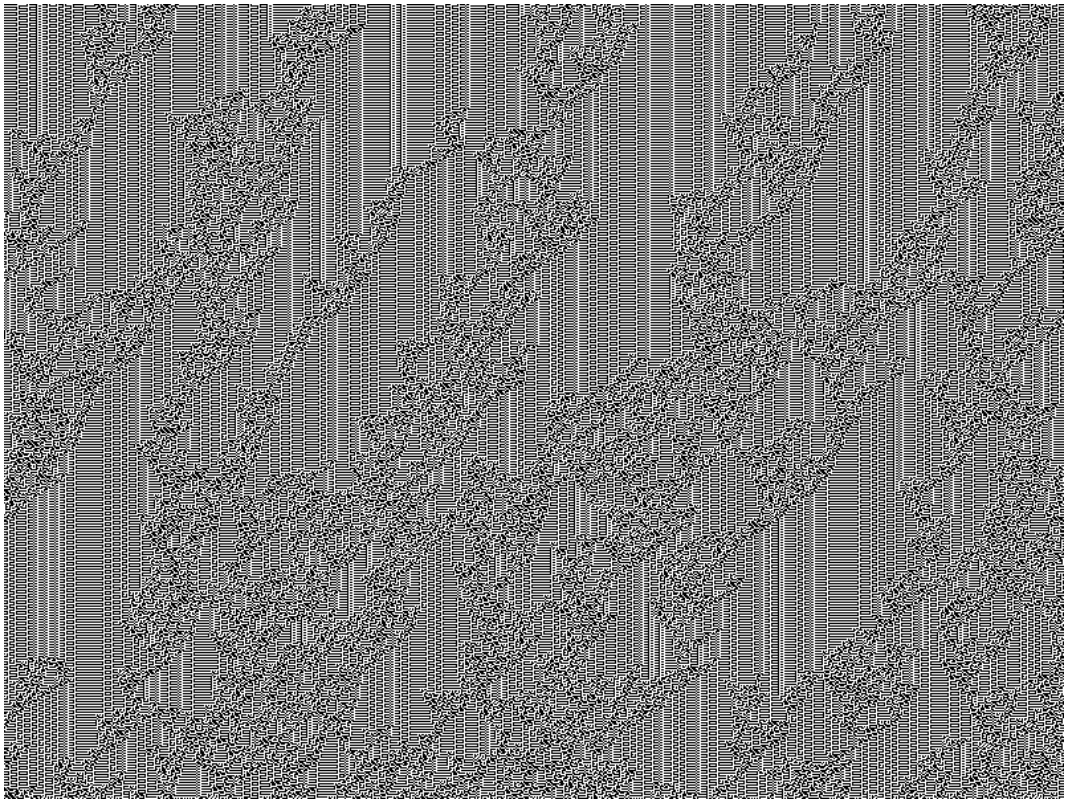
Les diagrammes espace-temps des automates cellulaires de classe 4 ont certains points communs. Une grande partie de l'espace-temps est rempli par un motif périodique de fond, une sorte d'éther à l'intérieur duquel des structures localisées, des sortes de particules ou de signaux, se déplacent et intéragissent au cours de leurs collisions, donnant naissance à de nouvelles particules. Des exemples de telles collisions sont représentés sur la figure 1.7.

De l'observation de ces collisions, on retient, plutôt que l'agencement des 2000 états qui composent chaque diagramme espace-temps, une vue schématique telle que celles représentées sur la figure 1.8. En effet, même si les images ne sont pas périodiques, elles sont localement très régulières : en dehors des collisions qui sont difficiles à décrire, les autres parties des diagrammes espace-temps sont composées de fond et de particules périodiques qui s'y déplacent selon une pente rationnelle.

► Il ne faut toutefois pas croire qu'on puisse passer à une vision continue des particules et des collisions et que les diagrammes espace-temps sont invariants par homothétie. En effet, dans le cas des automates cellulaires, les particules sont discrètes et possèdent donc une épaisseur. Ainsi, une collision n'est-elle pas nécessairement le lieu de rencontre de particules continues mais peut aussi être engendrée par une proximité de particules. D'autre part, les particules étant des motifs périodiques, la rencontre de deux particules peut engendrer plusieurs collisions différentes selon la synchronisation temporelle des particules au moment de la rencontre. Le modèle discret de collision de particules est fondamentalement différent de son analogue continu.

Ces considérations géométriques induisent une opération naturelle sur les diagrammes espace-temps des automates cellulaires complexes : une sorte d'opération de zoom arrière. L'intérêt de cette opération est qu'elle homogénéise le fond et qu'elle amincit les particules. La figure 1.9 présente une collision et la même collision après un zoom arrière (effectué grâce à un logiciel de traitement d'image).

Lorsqu'on applique une opération de zoom arrière sur un diagramme espace-temps, l'image obtenue n'est pas nécessairement le diagramme espace-temps d'un automate cellulaire. Aussi lui préfère-t-on une opération plus intéressante algébriquement et qui lui ressemble géométriquement : le groupage. Le groupage est un « zoom sans perte ». Comme pour un zoom, on découpe le diagramme espace-temps en carrés de côté k mais au lieu de remplacer chaque carré d'états par un état correspondant à une moyenne des états concernés, on le remplace par le k-uple des états de la ligne inférieure du carré. La figure 1.10 présente l'application d'une telle méthode à la figure précédente. La figure 1.11



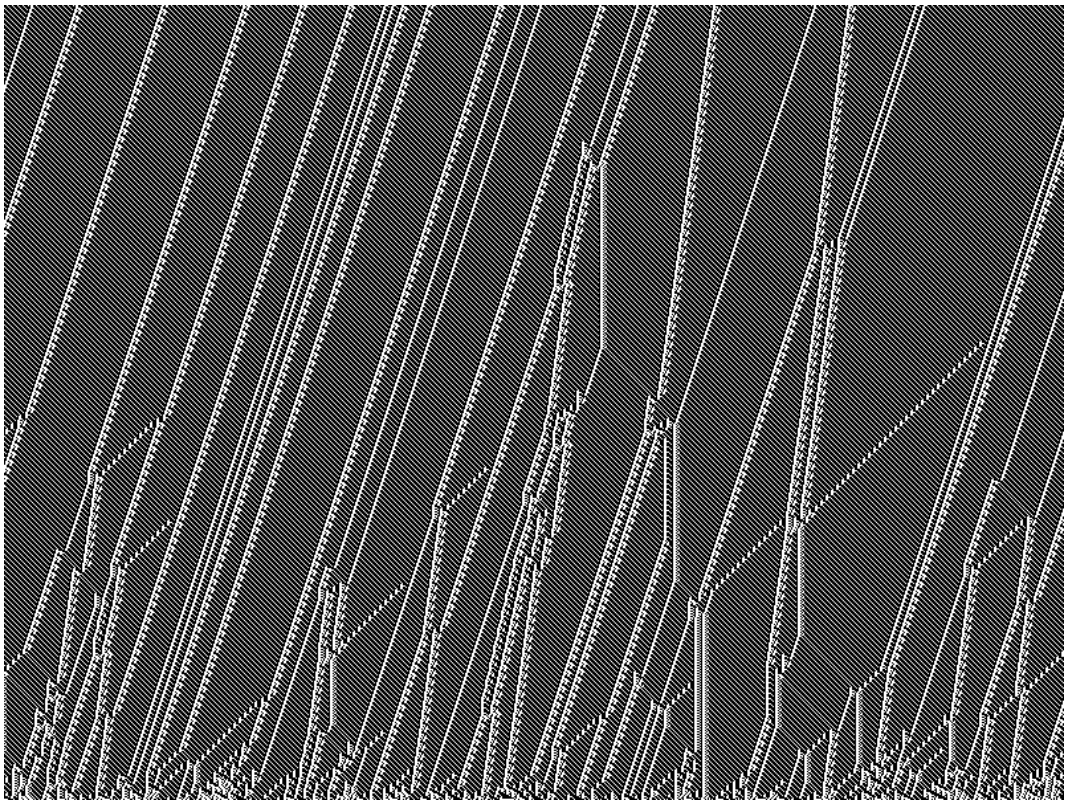
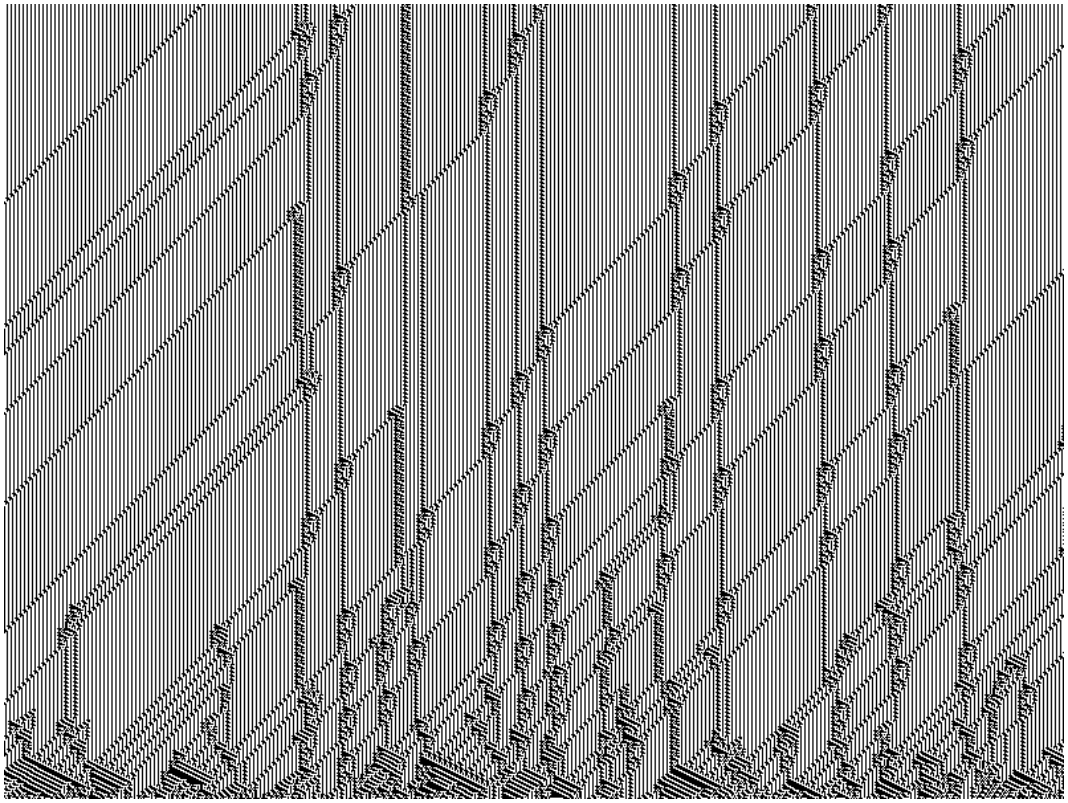


FIG. 1.6 – Diagrammes espace-temps d'automates cellulaires complexes (1/2)



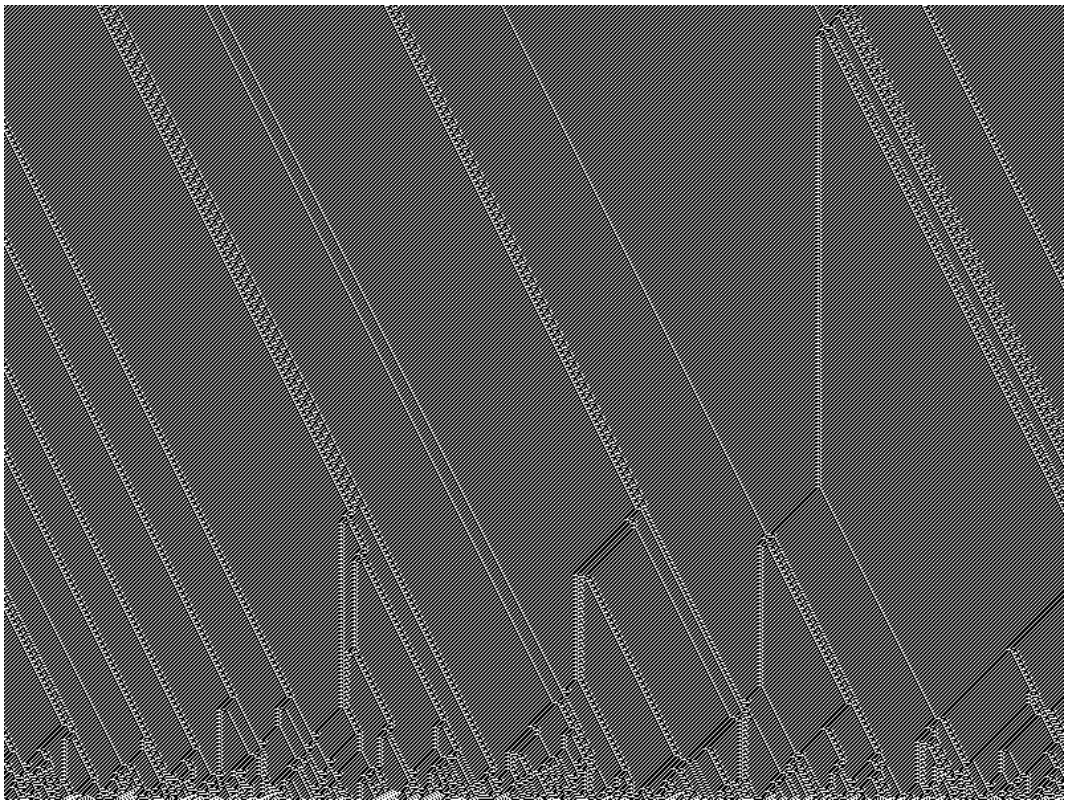


Fig. 1.6 – Diagrammes espace-temps d'automates cellulaires complexes (2/2)

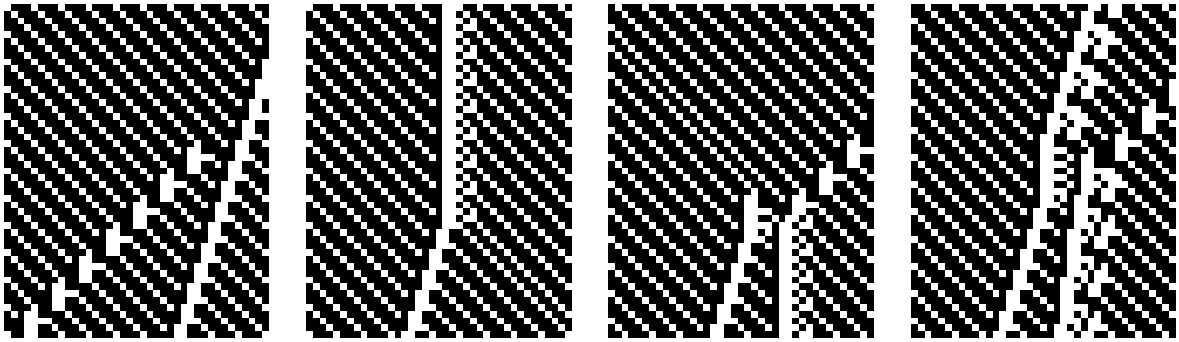


FIG. 1.7 – Automates cellulaires complexes : fonds, particules et collisions

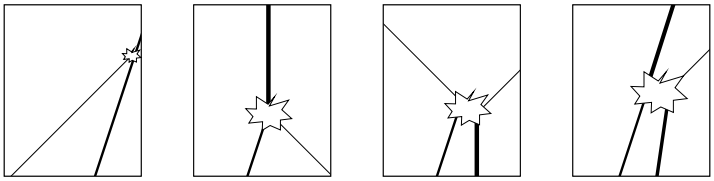


FIG. 1.8 – Automates cellulaires complexes : abstraction des collisions

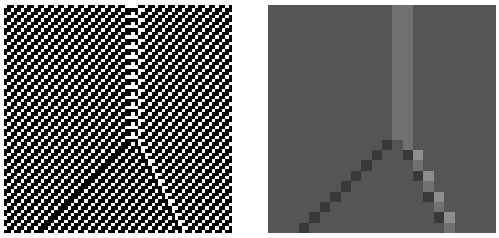


FIG. 1.9 – Opération de zoom arrière

présente le groupage des diagrammes espace-temps des automates cellulaires de classe 4 vus précédemment. L'utilisation de plusieurs couleurs pour désigner le fond correspond aux différentes synchronisations de ce dernier avec la grille de découpage des carrés.

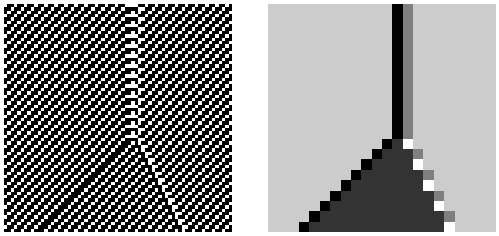


FIG. 1.10 – Opération de groupage

Algébriquement, la k-ième puissance $\mathcal{A}^k$ d'un automate cellulaire $\mathcal{A}$ à n états de rayon de voisinage r est un automate cellulaire de rayon de voisinage r à n^k états dont les diagrammes espace-temps sont mis en bijection avec les diagrammes espace-temps de $\mathcal{A}$ par l'opération de groupage par k. La règle locale de transition de $\mathcal{A}^k$ se construit aisément à partir de celle de $\mathcal{A}$.

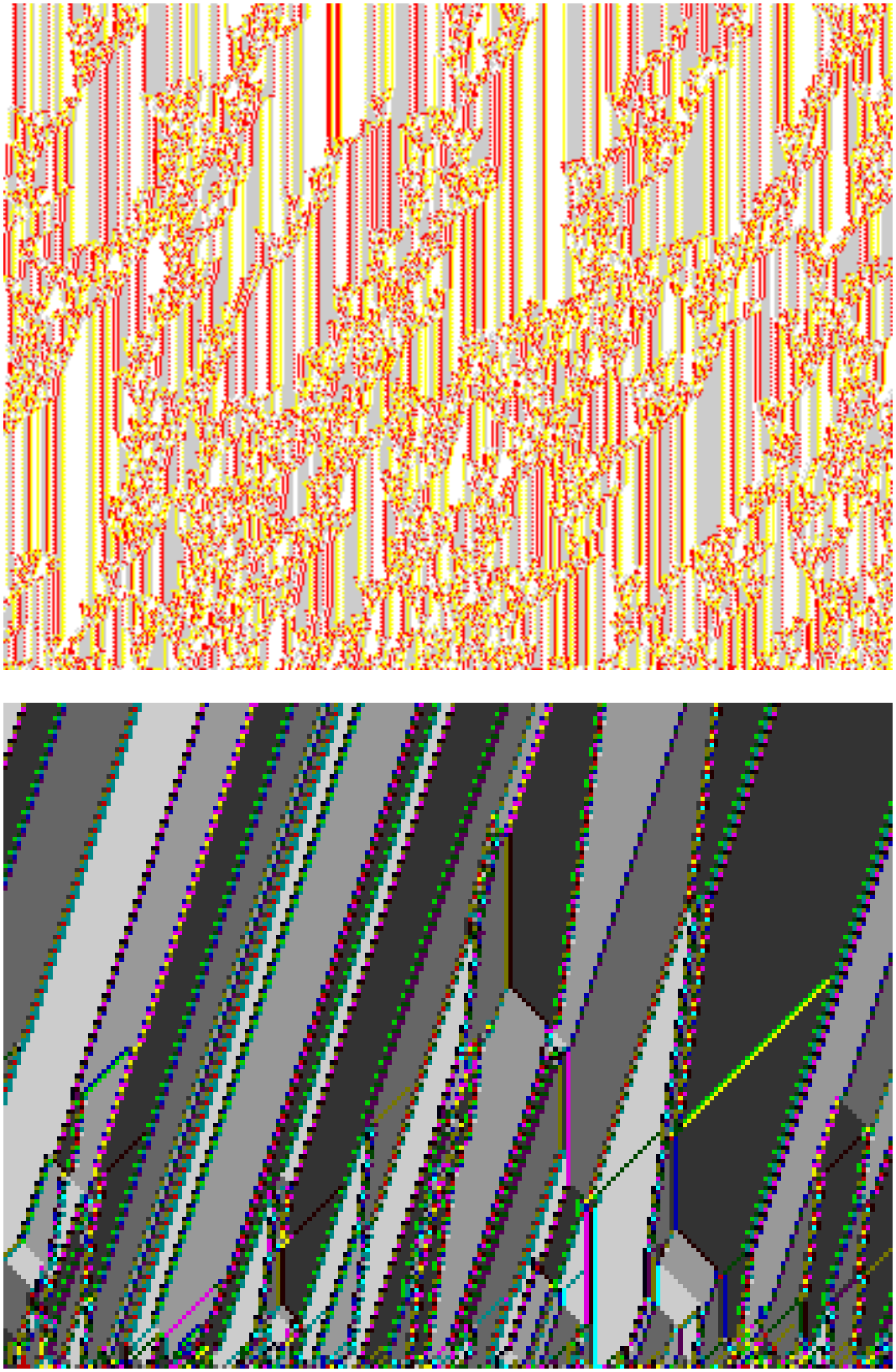


FIG. 1.11 – Diagrammes espace-temps d'automates cellulaires complexes groupés (1/2)

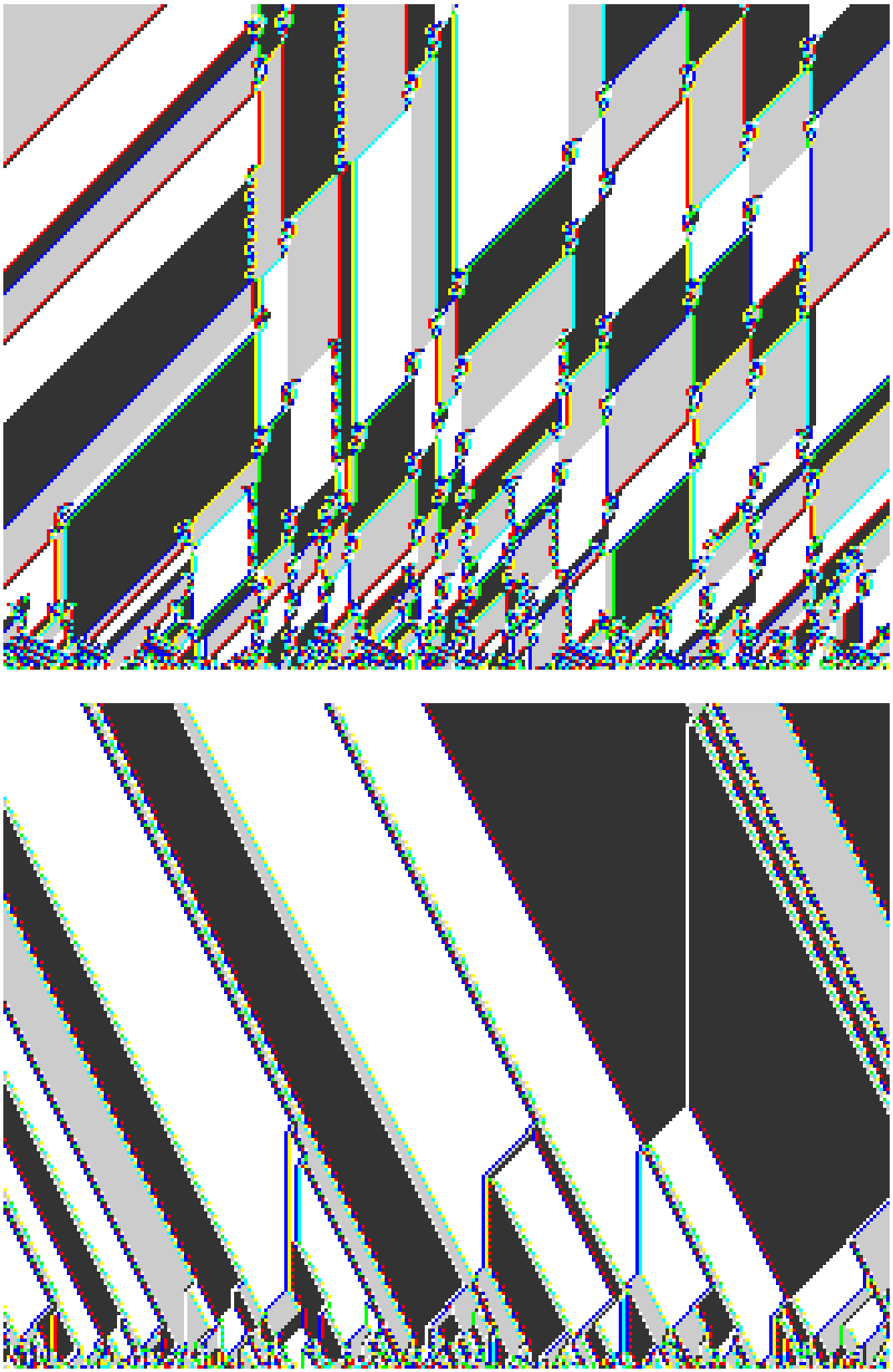


FIG. 1.11 – Diagrammes espace-temps d'automates cellulaires complexes groupés (2/2)

Une relation de préordre sur les automates cellulaires est alors définie de la manière suivante. Pour tout couple $\mathcal{A}$, $\mathcal{B}$ d'automates cellulaires, on pose $\mathcal{A} \leqslant_{\square} \mathcal{B}$ s'il existe deux puissances m et n telles que $\mathcal{A}^m \subseteq \mathcal{B}^n$.

Dans son travail de thèse, I. Rapaport [12] a introduit ce préordre et en a étudié certaines propriétés. Nous présentons ses travaux en détail en annexe afin de faciliter la mise en parallèle du groupage originel et de ses extensions ainsi que la comparaison des méthodes et outils utilisés pour les étudier.

1.2.2 Classification, calculabilité et complexité

Les classifications présentées ci-dessus reposent sur des notions de complexité variables : complexité visuelle des diagrammes espace-temps pour Wolfram, propriétés topologiques et de mesure ou encore propriétés géométriques et algébriques pour le groupage. La classification de K. Čulik II et S. Yu utilise un critère de décidabilité pour définir sa classe la plus haute. Nous proposons ici deux problèmes de décision qui nous semblent pertinents pour étudier la complexité algorithmique des automates cellulaires et nous discutons des limites de cette approche.

Indécidabilité de l'apparition d'un motif

Plutôt que l'accessibilité entre configurations finies proposée par K. Čulik II et S. Yu, nous proposons d'utiliser la famille de problèmes de décision CA-PAT. Par motif de dimension d de rayon r sur Q on entend une application de $[\![-r, r]\!]^d$ dans Q.

CA-PAT$_{\mathcal{A}}$	
Paramètre	un automate cellulaire $\mathcal{A}$
Entrée	une configuration initiale ultimement périodique C de $\mathcal{A}$ et un motif fini M sur $Q_{\mathcal{A}}$ de rayon r ;
Question	est-ce que le motif M apparaît dans l'évolution de $\mathcal{A}$ partant de C ?

Contrairement au précédent, ce problème est adapté à tous les automates cellulaires (avec ou sans état quiescent) et élude le problème de la croissance monotone des motifs non quiescents des configurations.

La complexité de CA-PAT$_{\mathcal{A}}$ varie avec $\mathcal{A}$ depuis une complexité linéaire en la taille du motif M et indépendante de C pour l'automate cellulaire constant, jusqu'à être indécidable, de même degré Turing que le problème de l'arrêt des machines de Turing, pour les automates cellulaires capables de simuler[4] des machines de Turing. La classe des automates cellulaires pour lesquels la complexité est de même degré Turing que le problème de l'arrêt correspond intuitivement aux automates cellulaires complexes au sens de la classe 4 de S. Wolfram : les interactions entre les particules sont si compliquées qu'elles sont imprévisibles.

[4]en un sens raisonnable... nous étudions en détail les problèmes liés à l'universalité et aux simulations dans le dernier chapitre.

Complexité de vérification d'un triangle de calcul

Plutôt que de s'intéresser à l'apparition d'un motif dans un diagramme espace-temps infini, on peut se demander quelle est la complexité de construction du diagramme espace-temps lui-même. Une manière de formaliser cette question est de considérer la famille de problèmes de décision CA-TRI.

$\mathsf{CA\text{-}TRI}_{\mathcal{A}}$	
Paramètre	un automate cellulaire $\mathcal{A}$ de rayon de voisinage r
Entrée	une durée T, un motif initial M de rayon rT et un état q de $Q_{\mathcal{A}}$;
Question	est-ce que le motif M évolue en q par $\mathcal{A}$ en temps T ?

La complexité en temps de $\mathsf{CA\text{-}TRI}_{\mathcal{A}}$ est bornée polynomialement en T car il suffit de calculer tout le triangle de calcul engendré par M. En dimension d, cela représente $\sum_{t=1}^{T-1}(2rt+1)^d$ applications de la règle locale de transition.

La complexité en temps de $\mathsf{CA\text{-}TRI}_{\mathcal{A}}$ varie avec $\mathcal{A}$ depuis une complexité constante pour l'automate cellulaire constant, en passant par une complexité linéaire en T pour les automates cellulaires périodiques, jusqu'à la P-complétude (pour une réduction raisonnable) pour les automates cellulaires les plus compliqués. La classe des automates cellulaires pour lesquels le problème est P-complet correspond intuitivement aux automates cellulaires complexes au sens de la classe 3 de S. Wolfram. La règle locale de transition mélange fortement l'information au point de rendre les évolutions très difficiles à prédire.

Limites de ces approches

Ces deux problèmes de décision pourraient certainement donner lieu à des classifications intéressantes. Cependant, le manque de connaissances sur les classes de complexité structurelle sous-polynomiale ou les degrés Turing récursivement énumérables limite la définition de classes pertinentes ou la démonstration de résultats élémentaires.

Le groupage, dans sa forme classique et dans les généralisations que nous en proposons, définit une sorte de réduction entre automates cellulaires qui possède de bonnes propriétés. En particulier, les deux problèmes de décision précédents sont compatibles avec le groupage, car – pour une réduction raisonnable – si un automate cellulaire est plus simple qu'un autre au sens du groupage alors le problème de décision CA-PAT, respectivement CA-TRI, du premier se réduit à celui du second.

CHAPITRE

2 GROUPAGES

On s'intéresse ici à la généralisation du groupage selon I. Rapaport, tel qu'il a été présenté au chapitre précédent. Après une présentation des objectifs de cette généralisation, un cadre formel est proposé pour étudier différents groupages. Une première tentative engendre une classification dont les automates cellulaires intrinsèquement universels sont un maximum global. Enfin, un raffinement de cette première proposition donne lieu à une classification du même type mais mieux structurée, la relation de préordre engendrant une structure de demi-treillis pour le produit cartésien. On s'intéresse alors aux idéaux de ce demi-treillis dont certains correspondent à des familles naturelles d'automates cellulaires.

Sommaire

2.1 Vers une extension du groupage . . . 48
2.1.1 Algorithmique . . . 48
2.1.2 Automates cellulaires unidirectionnels . . . 48
2.1.3 Automates cellulaires nilpotents . . . 49
2.1.4 Universalité intrinsèque . . . 49
2.1.5 Synthèse . . . 50
2.2 Interlude formel . . . 51
2.2.1 Axiomatisation . . . 51
2.2.2 Préordre . . . 54
2.2.3 Universalités . . . 55
2.3 Première esquisse du groupage . . . 56
2.3.1 Transformations géométriques . . . 57
Transformations de diagrammes espace-temps . . . 57
Trois transformations sympathiques . . . 58
Caractérisation des transformations sympathiques . . . 63
2.3.2 Axiomatisation . . . 67
Première tentative . . . 67
Deuxième tentative . . . 70
2.4 Une extension du groupage carré . . . 71
2.4.1 Définitions . . . 71
2.4.2 Espace des phases . . . 72
2.4.3 Exploration . . . 73
Liens avec le premier groupage . . . 73
Chaînes infinies incomparables . . . 75
Le bas de l'ordre . . . 76
Le haut de l'ordre . . . 77
Problèmes de décision associés . . . 79
2.4.4 Familles classiques d'automates cellulaires . . . 80
2.4.5 Structure . . . 81
2.5 Un groupage fortement structuré . . . 83
2.5.1 Transformations géométriques généralisées . . . 83
Caractérisation des transformations sympathiques . . . 84
2.5.2 Axiomatisation . . . 85

2.5.3 Exploration 85
Liens avec le groupage précédent 85
Demi-treillis et idéaux 85

2.1 Vers une extension du groupage

Le groupage, tel qu'il a été défini au chapitre précédent, est un outil original et intéressant pour étudier les évolutions des automates cellulaires. En particulier, certaines familles d'automates cellulaires intuitivement simples sont bien situées dans le bas de l'ordre et séparées les unes des autres. De plus, les groupages d'automates cellulaires algébriquement simples, comme les automates cellulaires linéaires sur $\mathbb{Z}_p$, sont simples à décrire algébriquement, ce qui permet leur étude. Cependant, le groupage ne prend pas en compte certaines transformations géométriques des diagrammes espace-temps raisonnables de la littérature que nous détaillons ci-dessous.

2.1.1 Algorithmique

La forme de changement d'échelle la plus courante dans les constructions algorithmiques sur automates cellulaires est une variante du groupage défini au chapitre précédent. Au lieu de conserver uniquement les états nécessaires au calcul des transitions – en dimension 1 il s'agit du segment inférieur du carré d'états – on conserve tous les états. On trouve ce type de changement d'échelle par exemple dans l'article sur la reconnaissance des nombres premiers par automates cellulaires de P.C. Fischer [18]. La figure 2.1 illustre le principe de cette transformation.

			$q_{c,6,1}$	$q_{c,6,2}$	$q_{c,6,3}$			
		$q_{l,5,3}$	$q_{c,5,1}$	$q_{c,5,2}$	$q_{c,5,3}$	$q_{r,5,1}$		
	$q_{l,4,2}$	$q_{l,4,3}$	$q_{c,4,1}$	$q_{c,4,2}$	$q_{c,4,3}$	$q_{r,4,1}$	$q_{r,4,2}$	
$q_{l,3,1}$	$q_{l,3,2}$	$q_{l,3,3}$	$q_{c,3,1}$	$q_{c,3,2}$	$q_{c,3,3}$	$q_{r,3,1}$	$q_{r,3,2}$	$q_{r,3,3}$
$q_{l,2,1}$	$q_{l,2,2}$	$q_{l,2,3}$	$q_{c,2,1}$	$q_{c,2,2}$	$q_{c,2,3}$	$q_{r,2,1}$	$q_{r,2,2}$	$q_{r,2,3}$
$q_{l,1,1}$	$q_{l,1,2}$	$q_{l,1,3}$	$q_{c,1,1}$	$q_{c,1,2}$	$q_{c,1,3}$	$q_{r,1,1}$	$q_{r,1,2}$	$q_{r,1,3}$

FIG. 2.1 – Transformation géométrique de P. C. Fischer

Ce type de transformation, très utile algorithmiquement, ne nous intéresse pas car, si le transformé d'un diagramme espace-temps est bien un diagramme espace-temps, lorsqu'on applique la transformation à l'ensemble des diagrammes espace-temps d'un automate cellulaire, on obtient un automate cellulaire partiel : toutes les configurations initiales possibles n'apparaissent pas parmi les diagrammes espace-temps transformés.

2.1.2 Automates cellulaires unidirectionnels

L'opération de changement d'échelle apparaît assez naturellement lorsqu'on s'intéresse aux différents voisinages et en particulier lorsqu'on étudie les liens entre voisinage de Moore et voisinage unidirectionnel.

L'approche de C. Choffrut et K. Čulik II [26] propose de simuler le comportement d'un automate cellulaire à voisinage de Moore par un automate cellulaire unidirectionnel qui effectue une transition en le double de temps et à une translation près comme l'illustre la figure 2.2.

FIG. 2.2 – Transformation géométrique de C. Choffrut et Čulik II

L'approche de O. H. Ibarra *et al.* [31] propose de simuler le comportement d'un automate cellulaire à voisinage de Moore par un automate cellulaire unidirectionnel qui regroupe plusieurs cellules en une seule, effectue les transitions en temps réel et à une translation près comme l'illustre la figure 2.3.

FIG. 2.3 – Transformation géométrique de O. H. Ibarra *et al.*

2.1.3 Automates cellulaires nilpotents

La nilpotence peut, elle aussi, être présentée en termes de changement d'échelle. Tout automate cellulaire nilpotent à n états de transitoire T peut être simulé par l'automate cellulaire nilpotent à deux états de transitoire 1 si on considère une transition sur T dans le premier automate cellulaire et des paquets d'au moins $k = \lceil \log_2 n \rceil$ états dans le second comme l'illustre la figure 2.4.

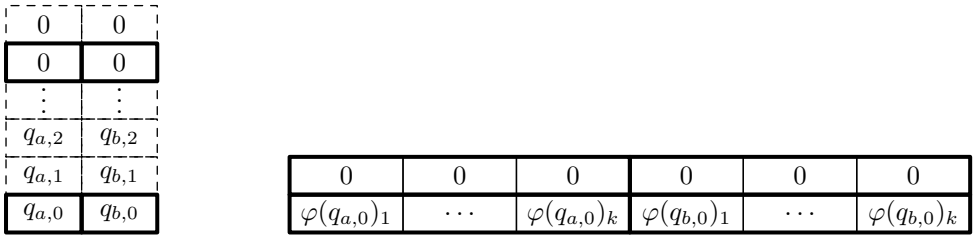


FIG. 2.4 – Transformation géométrique et nilpotence

2.1.4 Universalité intrinsèque

Enfin, on retrouve le changement d'échelle dans la notion d'universalité intrinsèque. Habituellement on désigne par automate cellulaire universel ce que nous appellerons automate cellulaire Turing universel ou universel pour le calcul. Ces automates cellulaires sont

capables, pour une notion de simulation raisonnable[1], de simuler tout calcul d'une machine de Turing. Le fait qu'un automate cellulaire soit Turing universel est obtenu de manière extrinsèque, en comparant des objets de natures différentes. Au contraire, les automates cellulaires intrinsèquement universels sont des automates cellulaires universels pour une notion intrinsèque d'universalité : ils simulent tous les autres automates cellulaires. En particulier, pour peu que les notions de simulation soient compatibles, l'universalité intrinsèque implique la Turing universalité. L'existence d'automates cellulaires intrinsèquement universels a commencé à apparaître dans les travaux d'E. R. Banks [20], puis des formalisations de plus en plus précises ont été proposées par J. Albert et K. Čulik II [34], par B. Martin [54] et par B. Durand et Zs. Róka [66].

La simulation retenue dans les articles ci-dessus est la suivante. Pour simuler un automate cellulaire avec un automate cellulaire intrinsèquement universel, on code chaque état de la configuration initiale par un segment d'états (ou un hyper-rectangle $\prod_{i=1}^{d} [\![0, m_i]\!]$ dans le cas de la dimension d) et une transition est simulée en un nombre constant de transitions. La figure 2.5 illustre ce type de simulation.

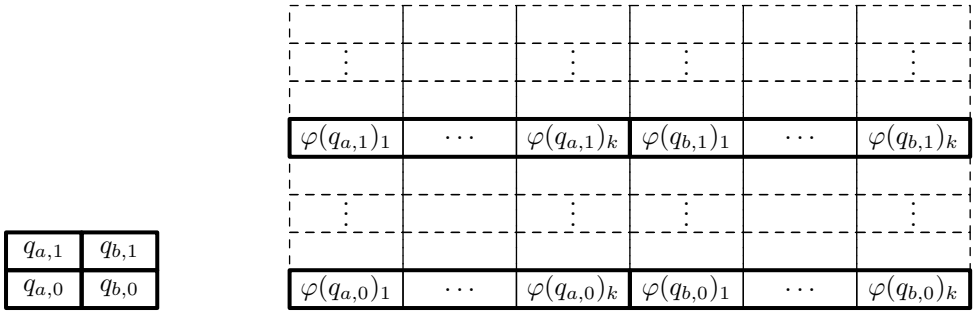


FIG. 2.5 – Transformation géométrique et universalité intrinsèque

Pour permettre à des automates cellulaires unidirectionnels d'être intrinsèquement universels, nous assouplirons cette définition en permettant éventuellement une translation régulière dans le temps. Nous verrons dans le chapitre suivant que cette nouvelle notion d'universalité intrinsèque est très robuste : si on essaie de l'assouplir plus, on obtient souvent encore la même famille d'automates cellulaires.

2.1.5 Synthèse

Nous avons observé trois types de transformations pour les automates cellulaires de dimension 1 : le découpage spatial en paquets, la translation régulière dans le temps et l'itération temporelle. Notons $\mathcal{A}^{\langle m,n,k\rangle}$ l'automate cellulaire transformé de l'automate cellulaire $\mathcal{A}$ par des paquets de taille m une itération temporelle de n et une translation régulière de k. Cet automate cellulaire est parfaitement défini et sa règle globale de transition peut se décomposer en

$$G_{\mathcal{A}}^{\langle m,n,k\rangle} = o_m^{-1} \circ \sigma_k \circ G_{\mathcal{A}}^{n} \circ o_m$$

où les différentes composantes sont définies ainsi :

[1] les différentes formes d'universalité sont discutées dans le dernier chapitre

o_m est l'application d'ouverture des paquets de taille m. C'est l'application bijective de $(Q_{\mathcal{A}}^m)^{\mathbb{Z}}$ dans $Q_{\mathcal{A}}^{\mathbb{Z}}$ dont l'inverse o_m^{-1} associe à une configuration C la configuration $o_m^{-1}(C)$ définie en toute position $p \in \mathbb{Z}$ par

$$o_m^{-1}(C)_p = \big(C_{mp}, \ldots, C_{(m+1)p-1}\big) \quad ;$$

σ_k est la translation de vecteur k sur $Q_{\mathcal{A}}$;

G^n est l'itéré n-ième de la règle globale de transition $G_{\mathcal{A}}$.

La figure 2.6 illustre ce type de transformation sur un diagramme espace-temps.

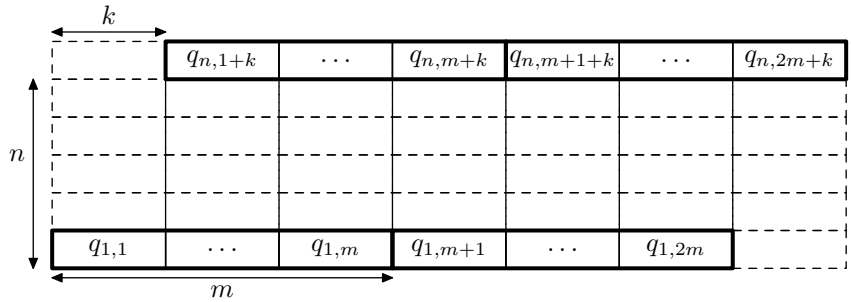


FIG. 2.6 – Transformation $\langle m, n, k\rangle$ d'un diagramme espace-temps

2.2 Interlude formel

Avant d'étudier les généralisations du groupage proprement dites, nous introduisons un formalisme logique abstrait par le biais d'une théorie. Ce système de notations permet d'extraire certaines propriétés structurelles importantes des objets manipulés et de leurs interactions avec l'opération de changement d'échelle et le préordre élémentaire choisis. Certaines démonstrations deviennent alors visuelles et relativement indépendantes du type de groupage considéré ce qui rend leur compréhension plus simple.

2.2.1 Axiomatisation

Nous proposons une formalisation reposant sur une théorie logique : une signature munie d'un jeu d'axiomes. Les différentes relations de simulations que nous définirons seront des modèles de cette théorie. En particulier, le groupage au sens d'I. Rapaport peut être considéré comme un tel modèle, nous l'utiliserons pour illustrer notre propos et nous le désignerons par l'expression groupage carré.

Définition 1. *Le* groupage abstrait $\mathfrak{A}$ *est une théorie logique de signature*

$$\begin{aligned}\big(\mathsf{Obj}, \mathsf{Trans};\ &\textbf{apply} : \mathsf{Obj} \times \mathsf{Trans} \to \mathsf{Obj},\\ &\textbf{divide} \subseteq \mathsf{Obj} \times \mathsf{Obj},\\ &\textbf{combine} : \mathsf{Trans} \times \mathsf{Trans} \to \mathsf{Trans}\big).\end{aligned}$$

Obj représente l'ensemble des objets à comparer. Par exemple, pour le groupage carré, l'ensemble des automates cellulaires de dimension 1 et de voisinage $[\![-1, 1]\!]$. Les objets seront notés par des lettres latines en bas de casse (*i.e.* x, y, z, ...)

Trans représente l'ensemble des transformations qu'on peut effectuer sur les objets. Par exemple, pour le groupage carré, l'ensemble $\mathbb{N}^*$ représentant les différentes puissances. Les transformations seront notées par des lettres grecques en bas de casse (*i.e.* α, β, γ, ...)

apply représente l'opération d'application d'une transformation à un objet. Par exemple, pour le groupage carré, il s'agit du passage à la puissance. Pour simplifier les écritures, on notera x^α au lieu de **apply**(x, α).

divide représente l'opération élémentaire de comparaison des objets. Par exemple, pour le groupage carré, il s'agit de la relation sous-automate. Pour simplifier les écritures, on notera $x \mid y$ au lieu de **divide**(x, y).

combine représente l'opération de composition des transformations. Par exemple, pour le groupage carré, il s'agit du produit sur $\mathbb{N}^*$. Pour simplifier les écritures, on notera $\alpha \cdot \beta$ au lieu de **combine**(α, β).

Pour visualiser les relations entre les objets dans ce formalisme, et pour simplifier les démonstrations, on utilise la représentation graphique suivante. On considère un graphe dont les sommets représentent les objets. La transformation d'un objet x par une transformation α (formellement x^α) est représentée par une flèche ondulée du sommet x au sommet x^α étiquetée par α. Le fait qu'un objet x divise un objet y (formellement $x \mid y$) est représenté par une flèche droite du sommet x au sommet y.

Dans ce formalisme, on dit qu'un objet x est simulé par un objet y, noté $x \preccurlyeq y$, si la formule $\exists\alpha\exists\beta\left(x^\alpha \mid y^\beta\right)$, notée $\phi(x, y)$, est satisfaisaite (*i.e.* $\mathfrak{A} \vdash \phi(x, y)$). La figure 2.7 représente graphiquement cette formule.

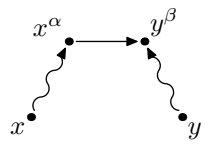


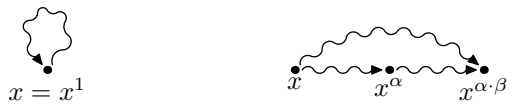
FIG. 2.7 – Simulation entre objets dans le groupage abstrait

On axiomatise cette théorie par des propriétés structurelles et des relations entre les différentes opérations. L'idée est de reprendre certaines propriétés du groupage carré, en les affaiblissant si nécessaires, pour obtenir une structure forte sur la relation de simulation.

Combinaison. La structure $(\mathsf{Trans}, \cdot)$ est un monoïde.

$$\begin{aligned} \mathfrak{A} \vdash \quad & \exists 1 \forall \alpha \left(\alpha \cdot 1 = \alpha \wedge 1 \cdot \alpha = \alpha\right) \\ \wedge \quad & \forall\alpha\forall\beta\forall\gamma \left((\alpha \cdot \beta) \cdot \gamma = \alpha \cdot (\beta \cdot \gamma)\right) \end{aligned}$$

Compatibilité. L'opérateur **apply** fait agir le monoïde $(\mathsf{Trans}, \cdot)$ sur l'ensemble Obj.

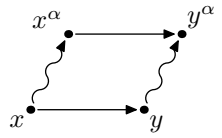


$$\mathfrak{A} \vdash \quad \forall x \left(x^1 = x\right) \quad \wedge \quad \forall x\forall\alpha\forall\beta \left(\left(x^\alpha\right)^\beta = x^{\alpha\cdot\beta}\right)$$

Divisibilité. La relation **divide** est une relation de préordre sur Obj.

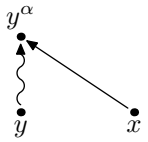
$$\mathfrak{A} \vdash \quad \forall x\,(x \mid x) \quad \wedge \quad \forall x \forall y \forall z\,((x \mid y \wedge y \mid z) \to x \mid z)$$

Transitivité. L'opérateur **apply** est compatible avec la relation **divide**.



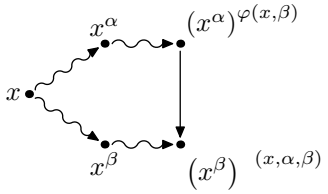
$$\mathfrak{A} \vdash \quad \forall x \forall y \forall \alpha\,(x \mid y \to x^\alpha \mid y^\alpha)$$

Surjectivité. l'opérateur **apply** préserve la richesse des objets.



$$\mathfrak{A} \vdash \quad \forall \alpha \forall x \exists y\,(x \mid y^\alpha)$$

Proximité. L'opérateur **apply** n'éloigne pas trop les transformations d'un même objet. C'est-à-dire qu'il existe deux fonctions φ et ψ telles que



$$\mathfrak{A} \vdash \quad \forall x \forall \alpha \forall \beta \left((x^\alpha)^{\varphi(x,\beta)} \mid \left(x^\beta\right)^{\psi(x,\alpha,\beta)} \right)$$

En toute rigueur, il faudrait ajouter les fonctions φ et ψ à la signature du groupage abstrait. Nous ne le ferons pas explicitement pour ne pas encombrer les notations.

- Cet axiome peut sembler, *a priori*, moins naturel que les précédents. En fait, dans le cas où le monoïde des transformations est commutatif, on peut poser $\varphi(x,\beta) = \beta$ et $\psi(x,\alpha,\beta) = \alpha$. Dans le cas où ce monoïde n'est pas commutatif, l'axiome sert de version faible à la commutativité. Si la relation de simulation est un préordre, alors les axiomes précédents impliquent une version faible de l'axiome de **Proximité** :

$$\mathfrak{A} \vdash \quad \forall x \forall \alpha \forall \beta \exists \gamma \exists \delta \left((x^\alpha)^\gamma \mid \left(x^\beta\right)^\delta \right)$$

  Nous verrons au paragraphe suivant qu'il y a en fait équivalence entre ces deux propriétés. L'utilisation d'une version plus forte, l'axiome de Proximité, permet de montrer un résultat intéressant concernant les objets maximaux pour l'ordre induit.

Nous pouvons maintenant étudier quelques propriétés élémentaires de cette théorie. Une propriété dérivable dans cette théorie, c'est-à-dire une formule φ telle que $\mathfrak{A} \vdash \varphi$, est appelée une *propriété de groupage*.

2.2.2 Préordre

Théorème 11. *« $\preccurlyeq$ est une relation de préordre » est une propriété de groupage.*

$$\mathfrak{A} \vdash \quad \forall x\,(x \preccurlyeq x) \quad \wedge \quad \forall x \forall y \forall z\,((x \preccurlyeq y \wedge y \preccurlyeq z) \rightarrow x \preccurlyeq z)$$

Démonstration. La démonstration est en deux parties. Tout d'abord, on montre que la relation est réflexive.

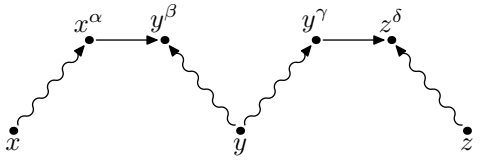
$$\mathfrak{A} \vdash \quad \forall x\,(x \preccurlyeq x)$$

C'est une conséquence directe des axiomes de **Compatibilité** et de **Divisibilité**.

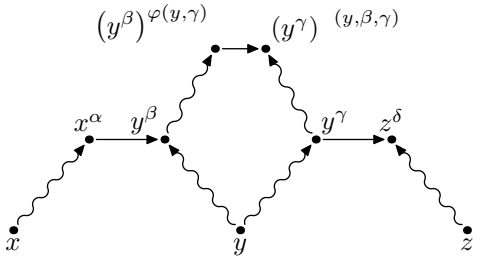
Ensuite, on vérifie que la relation est transitive.

$$\mathfrak{A} \vdash \quad \forall x \forall y \forall z\,((x \preccurlyeq y \wedge y \preccurlyeq z) \rightarrow x \preccurlyeq z)$$

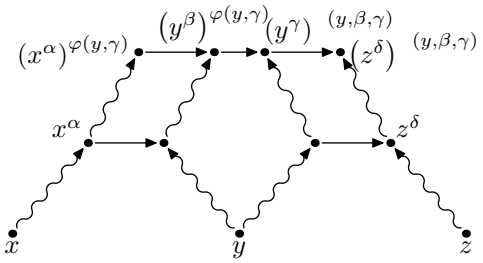
Soient x, y et z trois éléments de Obj et α, β, γ et δ quatre éléments de Trans tels que $x^\alpha \mid y^\beta$ et $y^\gamma \mid z^\delta$. On va exhiber deux éléments ν et ϵ de Trans vérifiant $x^\nu \mid z^\epsilon$.



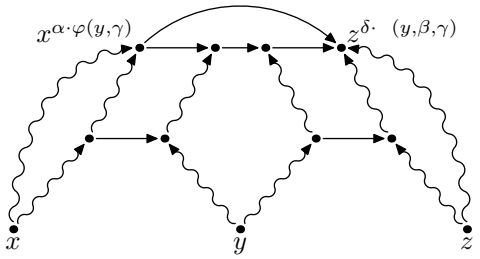
On applique l'axiome de **Proximité** avec pour paramètres y, β et γ. On obtient alors $\left(y^\beta\right)^{\varphi(y,\gamma)} \mid (y^\gamma)^{\psi(y,\beta,\gamma)}$.



En appliquant deux fois l'axiome de **Transitivité**, à gauche avec pour paramètres x^α, y^β et $\varphi(y,\gamma)$ et à droite avec les paramètres y^γ, z^δ et $\psi(y,\beta,\gamma)$, on obtient deux nouvelles relations : $(x^\alpha)^{\varphi(y,\gamma)} \mid \left(y^\beta\right)^{\varphi(y,\gamma)}$ et $(y^\gamma)^{\psi(y,\beta,\gamma)} \mid \left(z^\delta\right)^{\psi(y,\beta,\gamma)}$.



De l'axiome de **Divisibilité**, on déduit $(x^\alpha)^{\varphi(y,\gamma)} \mid \left(z^\delta\right)^{\psi(y,\beta,\gamma)}$. Utilisant l'axiome de **Compatibilité**, on obtient alors $x^{\alpha\cdot\varphi(y,\gamma)} \mid z^{\delta\cdot\psi(y,\beta,\gamma)}$. Ainsi, $\nu = \alpha \cdot \varphi(y,\gamma)$ et $\epsilon = \delta \cdot \psi(y,\beta,\gamma)$ sont des choix acceptables.



■

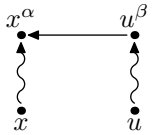
2.2.3 Universalités

Nous distinguons deux sortes d'universalité intrinsèque. Dans la version faible, un objet u est *universel* s'il peut simuler tous les objets, formellement $\forall x\,(x \preccurlyeq u)$, c'est-à-dire $\forall x \exists \alpha \exists \beta \left(x^\alpha \mid u^\beta\right)$. Dans la version forte, un objet u est *fortement universel* s'il peut simuler tous les objets mais de manière plus directe, formellement $\forall x \exists \alpha\,(x \mid u^\alpha)$.

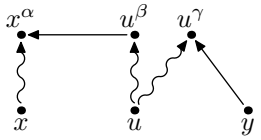
Théorème 12. *« S'il existe un objet fortement universel, alors tous les objets universels sont fortement universels » est une propriété de groupage.*

$$\mathfrak{A} \vdash \exists u \forall x \exists \alpha\,(x \mid u^\alpha) \to \forall x \left(\forall y\,(y \preccurlyeq x) \to \forall y \exists \beta \left(y \mid x^\beta\right)\right)$$

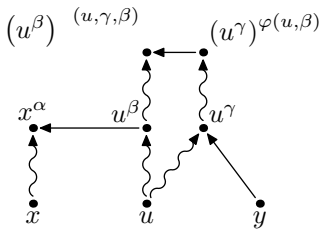
Démonstration. Soit u un objet fortement universel et x un objet universel. Montrons que x est fortement universel. Soient α et β deux éléments de Trans vérifiant $u^\beta \mid x^\alpha$.



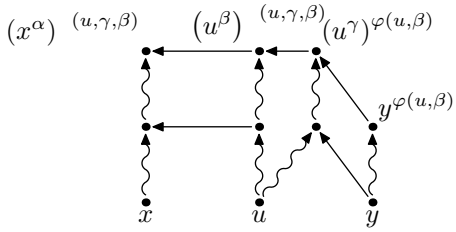
Soit y un élément quelconque de Obj. Comme u est fortement universel, il existe γ tel que $y \mid u^\gamma$.



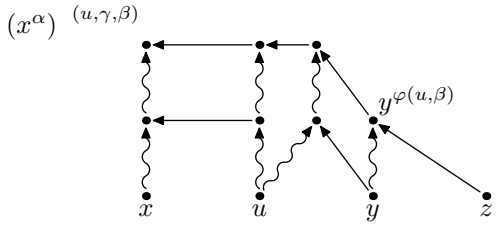
On applique l'axiome de **Proximité** avec pour paramètres u, β et γ. On obtient la relation $(u^\gamma)^{\varphi(u,\beta)} \mid \left(u^\beta\right)^{\psi(u,\gamma,\beta)}$.



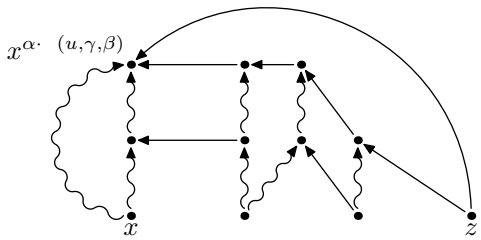
En appliquant deux fois l'axiome de **Transitivité**, à gauche avec pour paramètres x^α, u^β et $\psi(u,\gamma,\beta)$ et à droite avec pour paramètres u^γ, y et $\varphi(u,\beta)$, on obtient deux nouvelles relations : $\left(u^\beta\right)^{\psi(u,\gamma,\beta)} \mid (x^\alpha)^{\psi(u,\gamma,\beta)}$ et $y^{\varphi(u,\beta)} \mid (u^\gamma)^{\varphi(u,\beta)}$.



Observons que $\varphi(u,\beta)$ ne dépend que de u et de x. Par application de l'axiome de **Surjectivité**, pour tout élément z de Obj, on peut choisir un élément y de Obj tel que les équations précédentes restent toutes valides et qu'en plus on ait $z \mid y^{\varphi(u,\beta)}$.



De l'axiome de **Divisibilité**, on déduit que pour tout élément z de Obj, on a $z \mid (x^\alpha)^{\psi(u,\gamma,\beta)}$. Par l'axiome de **Compatibilité**, on peut alors conclure $z \mid x^{\alpha\cdot\psi(u,\gamma,\beta)}$.



■

2.3 Première esquisse du groupage

Nous proposons une première généralisation du groupage. L'ensemble des objets considérés est maintenant l'ensemble des automates cellulaires de dimension d et non plus uniquement ceux de rayon de voisinage 1 car nous cherchons à nous libérer de la représentation syntaxique de la règle locale de transition. Le préordre élémentaire reste la relation sous-automate car elle semble pertinente, au vu de ses propriétés algébriques présentées dans le premier chapitre de cette thèse, ainsi que pour son interprétation sur l'ensemble des diagrammes espace-temps. En ce qui concerne les transformations permises sur les automates cellulaires, nous sommes intéressés par des transformations de nature géométrique sur les diagrammes espace-temps et qui aient un sens algébrique sur les règles globales de transitions. Nous formalisons cette idée de transformation géométrique, caractérisons les transformations intéressantes et étudions le groupage engendré.

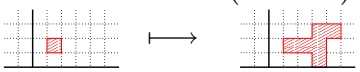
2.3.1 Transformations géométriques

Transformations de diagrammes espace-temps

Transformation. Intuitivement, une transformation de nature géométrique sur les diagrammes espace-temps est une transformation qui ne tient pas compte directement de l'ensemble d'états de l'automate cellulaire transformé. La même transformation appliquée à plusieurs diagrammes espace-temps engendre des diagrammes espace-temps où l'état d'une cellule à un temps et une position donnés est simplement un uple d'états de cellules du diagramme espace-temps d'origine. Se donner une transformation géométrique c'est donc se donner une application de l'espace-temps d'arrivée dans l'ensemble des uples de l'espace-temps de départ, qui à une cellule du diagramme espace-temps destination associe le uple de cellules du diagramme espace-temps d'origine dont les états constituent l'état de la cellule destination. Nous proposons la formalisation suivante.

Définition 2. *Une* transformation géométrique *est une paire* $(k, \mathbf{\Lambda})$ *où* k *est un entier strictement positif et* $\mathbf{\Lambda}$ *une application*

$$\mathbf{\Lambda} : \quad \mathbb{N} \times \mathbb{Z}^d \longrightarrow \left(\mathbb{N} \times \mathbb{Z}^d\right)^k$$



► On peut s'interroger sur la généralité de cette définition, car on a fixé un même k pour toutes les cellules. De notre point de vue, ce choix peut se justifier de deux manières. Tout d'abord, la formalisation est déjà suffisamment complexe à k fixé pour qu'on ne la complique pas d'avantage. D'autre part, si on laisse k libre, les transformations sympathiques engendrées (au sens défini formellement plus loin) seront toutes à k fixé. Aussi, nous avons jugé pertinent de fixer k dès la définition des transformations géométriques.

Mise en oeuvre. La transformation de diagrammes espace-temps sur l'alphabet Q par une transformation géométrique $(k, \mathbf{\Lambda})$ est définie par l'application $\overline{\mathbf{\Lambda}}_Q$ de $Q^{\mathbb{N}\times\mathbb{Z}^d}$ dans $\left(Q^k\right)^{\mathbb{N}\times\mathbb{Z}^d}$ définie pour tout diagramme espace-temps Δ sur Q, en tout temps $t \in \mathbb{N}$ et en toute position $p \in \mathbb{Z}^d$ par $\overline{\mathbf{\Lambda}}_Q(\Delta)(t,p) = (\Delta(\mathbf{\Lambda}(t,p)_1), \ldots, \Delta(\mathbf{\Lambda}(t,p)_k))$.

Exemple. En dimension 1, la transformation $(3, \mathbf{\Lambda}^{\langle 3,4,1\rangle})$ définie en tout temps t et en toute position p par

$$\mathbf{\Lambda}^{\langle 3,4,1\rangle}(t,p) = ((4t, 3p+t), (4t, 3p+t+1), (4t, 3p+t+2))$$

transforme tout automate cellulaire $\mathcal{A}$ en l'automate cellulaire $\mathcal{A}^{\langle 3,4,1\rangle}$. La figure 2.8 illustre cette transformation.

Pseudo-composition. On définit la composition de deux transformations géométriques en observant l'application successive de ces transformations sur un diagramme espace-temps. Soient $(k, \mathbf{\Lambda})$ et $(k', \mathbf{\Lambda}')$ deux transformations géométriques. L'application successive de ces deux transformations engendre des diagrammes espace-temps sur $\left(Q^k\right)^{k'}$. Puisqu'on travaille toujours à un renommage des états près, on peut considérer ces diagrammes espace-temps comme étant à valeurs dans $Q^{kk'}$ et étant engendrés par application de la transformation $(kk', \mathbf{\Lambda}' \circ \mathbf{\Lambda})$ définie en tout temps $t \in \mathbb{N}$ et en toute position $p \in \mathbb{Z}^d$ par

$$(\mathbf{\Lambda}' \circ \mathbf{\Lambda})(t,p) = \left(\mathbf{\Lambda}\left(\mathbf{\Lambda}'(t,p)_1\right)_1, \ldots, \mathbf{\Lambda}\left(\mathbf{\Lambda}'(t,p)_{k'}\right)_k\right) \quad .$$

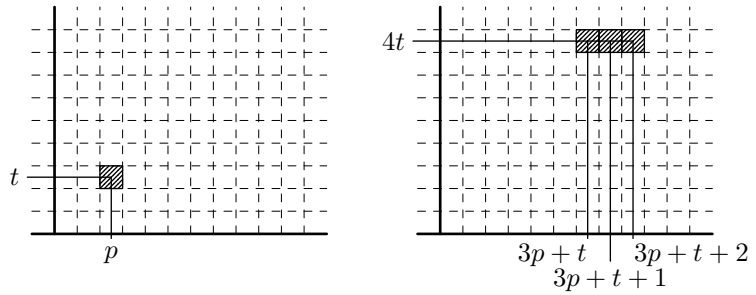


FIG. 2.8 – Exemple de transformation géométrique : $\left(3, \mathbf{\Lambda}^{\langle 3,4,1\rangle}\right)$

- Les transformations géométriques étant définies dans le sens inverse de leur application (à une cellule du diagramme espace-temps transformé on associe un uple de cellules du diagramme espace-temps de départ), la composition de $\mathbf{\Lambda}$ par $\mathbf{\Lambda}'$ applique $\mathbf{\Lambda}$ sur l'image d'une cellule par $\mathbf{\Lambda}'$ et non l'inverse.

L'opération correspondant – à un renommage près – à la composition sur l'application des tranformations, la notation composition est pertinente et a toutes les propriétés de la composition, en particulier l'associativité.

Pré-images. On introduit l'application $\widetilde{\mathbf{\Lambda}}$, qui à un ensemble de cellules d'un diagramme espace-temps associe l'ensemble des cellules qu'elles représentent.

$$\begin{array}{rccl} \widetilde{\mathbf{\Lambda}} : & 2^{\mathbb{N}\times\mathbb{Z}^d} & \longrightarrow & 2^{\mathbb{N}\times\mathbb{Z}^d} \\ & X & \longmapsto & \displaystyle\bigcup_{(t,p)\in X} \{\mathbf{\Lambda}(t,p)_1, \ldots, \mathbf{\Lambda}(t,p)_k\} \end{array}$$

Transformations sympathiques. On définit maintenant les transformations intéressantes dans le cadre du groupage. Ces transformations ont deux contraintes : la première garantit que l'application d'une transformation à l'ensemble des diagrammes espace-temps d'un automate cellulaire définit bien un automate cellulaire transformé, via l'ensemble complet des ses diagrammes espace-temps ; la seconde contrainte est purement technique et sert à éliminer des cas dégénérés.

Définition 3. *Une transformation géométrique* $(k, \mathbf{\Lambda})$ *est une* transformation géométrique sympathique *si elle vérifie :*

1. *pour tout automate cellulaire* $\mathcal{A}$*, il existe un automate cellulaire* $\mathcal{B}$ *vérifiant*

$$\left\{\overline{\mathbf{\Lambda}}_{Q_{\mathcal{A}}}(\Delta)\right\}_{\Delta\in\mathrm{Diag}(\mathcal{A})} = \mathrm{Diag}(\mathcal{B})$$

2. *pour tout temps* $t \in \mathbb{N}$*,* $\widetilde{\mathbf{\Lambda}}(\{t+1\}\times\mathbb{Z}^d) \not\subseteq \widetilde{\mathbf{\Lambda}}(\{t\}\times\mathbb{Z}^d)$*.*

Dans la suite, on appellera automate cellulaire transformé de l'automate cellulaire $\mathcal{A}$, par la transformation sympathique $(k, \mathbf{\Lambda})$, l'automate cellulaire $\mathcal{B}$ dont l'ensemble des diagrammes espace-temps est l'ensemble des diagrammes espace-temps de $\mathcal{A}$ transformés par $(k, \mathbf{\Lambda})$.

Trois transformations sympathiques

Parmi les transformations sympathiques, il existe trois familles de transformations particulières. Ces transformations s'expriment simplement en terme de composition sur

les règles globales de transition et correspondent aux opérations algorithmiques présentées au début du chapitre.

Notations. Afin de simplifier les écritures, on introduit tout d'abord quelques opérateurs sur $\mathbb{N} \times \mathbb{Z}^d$ et $\left(\mathbb{N} \times \mathbb{Z}^d\right)^k$. Certaines de ces opérations sont des opérations standard d'algèbre linéaire. La figure 2.9 illustre ces opérations.

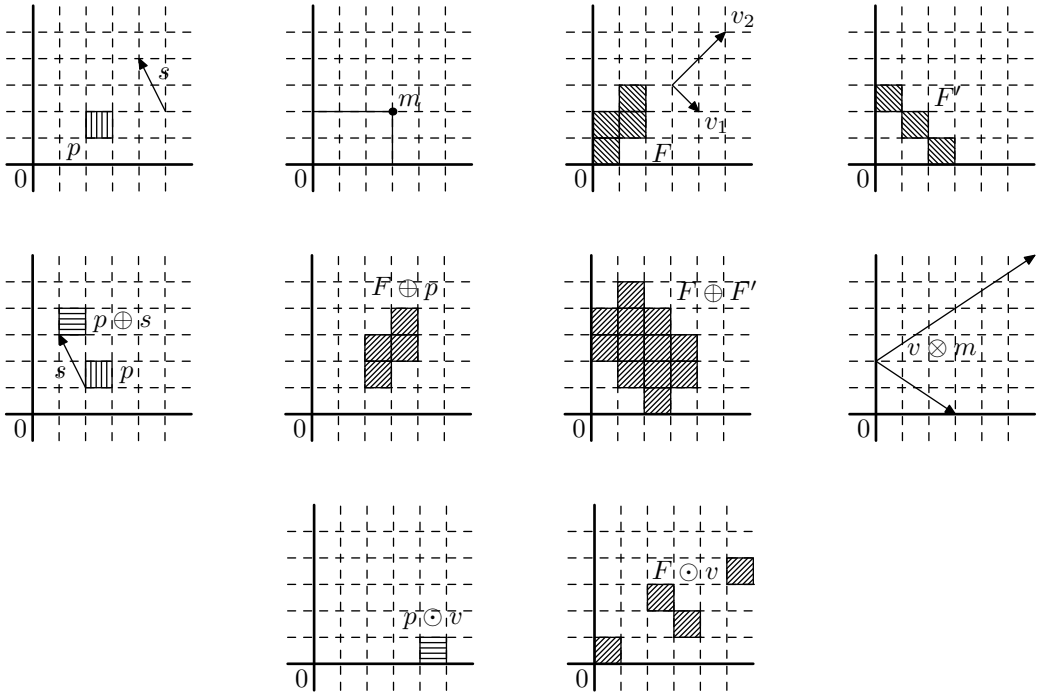


FIG. 2.9 – Opérateurs géométriques

L'opérateur $\circledast : \mathbb{N} \times \left(\mathbb{Z}^d\right)^k \to \left(\mathbb{N} \times \mathbb{Z}^d\right)^k$ associe à tout temps t et tout k-uples de positions F le k-uples de positions spatio-temporelles

$$t \circledast F = ((t, F_1), (t, F_2), \ldots, (t, F_k)) \quad .$$

L'opérateur $\oplus : \mathbb{Z}^d \times \mathbb{Z}^d \to \mathbb{Z}^d$ associe à toute position p et à tout vecteur s de $\mathbb{Z}^d$ la position

$$p \oplus s = (p_1 + s_1, p_2 + s_2, \ldots, p_k + s_k) \quad .$$

Cet opérateur s'étend naturellement aux k-uples de positions F par

$$F \oplus p = (F_1 \oplus p, F_2 \oplus p, \ldots, F_k \oplus p) \quad .$$

Enfin il s'étend naturellement aux couples de k-uple de positions F et k'-uple de positions F' par

$$F \oplus F' = (F_1 \oplus F'_1, F_2 \oplus F'_1, \ldots, F_k \oplus F'_{k'}) \quad .$$

L'opérateur $\ominus : \mathbb{Z}^d \times \mathbb{Z}^d \to \mathbb{Z}^d$ est défini comme $\oplus$ en remplaçant l'addition par la soustraction. Pour toute position p et tout vecteur s de $\mathbb{Z}^d$, on a

$$p \ominus s = (p_1 - s_1, p_2 - s_2, \ldots, p_k - s_k) \quad .$$

L'opérateur $\odot : \mathbb{Z}^d \times \left(\mathbb{Z}^d\right)^d \to \mathbb{Z}^d$ associe à toute position p et à toute famille libre de d vecteurs v de $\mathbb{Z}^d$ la position

$$p \odot v = p_1 v_1 \oplus p_2 v_2 \oplus \cdots \oplus p_d v_d \quad .$$

Cet opérateur s'étend naturellement aux k-uples de positions F par

$$F \odot v = (F_1 \odot v, F_2 \odot v, \ldots, F_k \odot v) \quad .$$

En particulier, pour toute paire de familles libres de d vecteurs v et v' de $\mathbb{Z}^d$ on a

$$v \odot v' = (v_1 \odot v', v_2 \odot v', \ldots, v_d \odot v') \quad .$$

L'opérateur $\otimes : \left(\mathbb{Z}^d\right)^d \times \mathbb{Z}^d \to \left(\mathbb{Z}^d\right)^d$ associe à toute famille libre de vecteurs v de $\mathbb{Z}^d$ et à tout vecteur m de $\mathbb{Z}^d$ la famille de vecteurs

$$v \otimes m = v \odot (m_1 \varsigma_1, \ldots, m_d \varsigma_d)$$

où ς_i désigne le vecteur dont toutes les coordonnées sont égales à 0 sauf la i-ème qui est égale à 1.

Découpage spatial. Une transformation purement spatiale consiste à plonger l'espace dans lui-même en découpant l'espace régulièrement à l'aide d'un k-uple $F \in \left(\mathbb{Z}^d\right)^k$ et en munissant ce pavage du plan d'une famille libre de vecteurs $v \in \left(\mathbb{Z}^d\right)^d$ telle que les translations de F par v recouvrent chaque case de $\mathbb{Z}^d$ une fois et une seule. Formellement, si $\mathbf{\Theta}_{F,v}$ est l'application qui à une cellule transformée associe les cellules originales de l'espace, on a

$$\mathbf{\Theta}_{F,v}(p) = F \oplus p \odot v \quad .$$

▶ Les paires (F, v) valides se caractérisent aisément. Pour toute famille libre v de vecteurs de $\mathbb{Z}^d$, il existe des uples F acceptables. Soit V le réseau associé à v, *i.e.* $V = \mathbb{Z}^d \odot v$. On définit une relation d'équivalence $\equiv_v$ pour toute paire p_1, p_2 de points de $\mathbb{Z}^d$ par $p_1 \equiv_v p_2$ si et seulement $p_1 \ominus p_2 \in V$. Cette relation induit exactement $k = \det(v)$ classes d'équivalence. Les uples F acceptables sont k-uples de points de $\mathbb{Z}^d$ tels que chaque point appartient à une classe d'équivalence distincte. La figure 2.10 présente un exemple de réseau associé à une famille libre de vecteurs ainsi que le choix d'un uple à l'aide des classes d'équivalence associées.

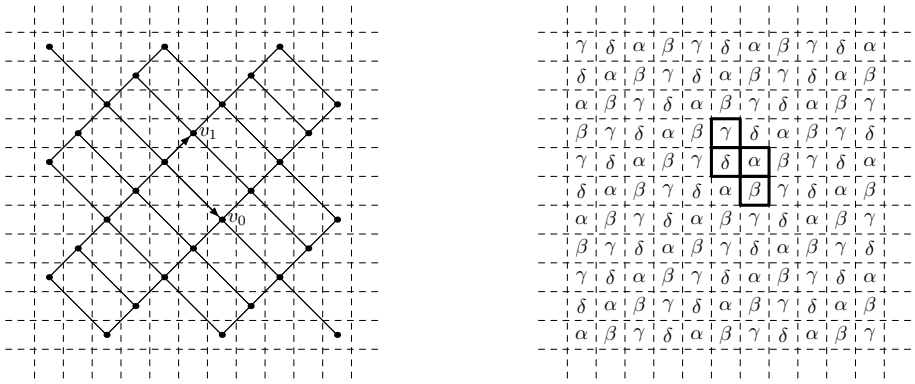


FIG. 2.10 – Réseau associé à une famille de vecteurs de $\mathbb{Z}^2$ et pavage

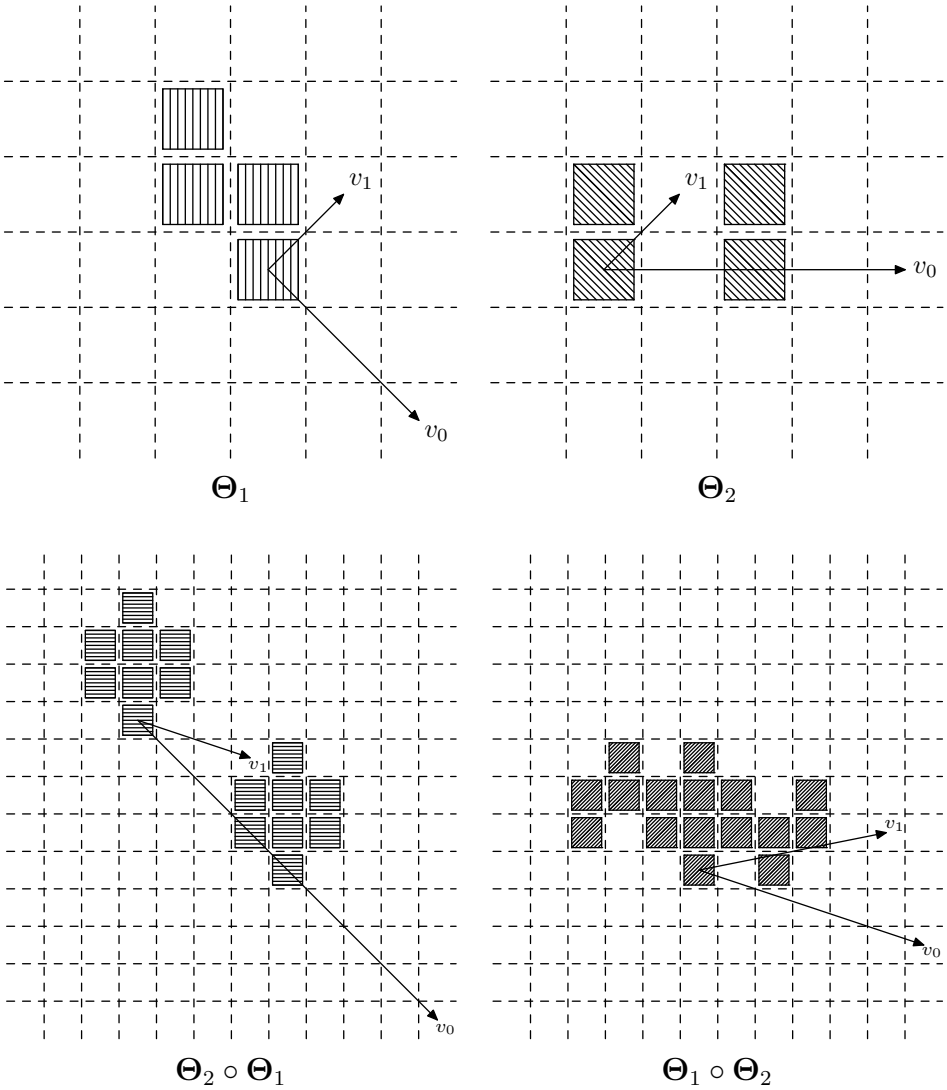


FIG. 2.11 – Composition de plongements de $\mathbb{Z}^2$ dans lui-même

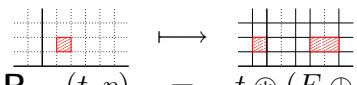
La composition de deux telles transformations est aussi une transformation de ce type :

$$\mathbf{\Theta}_{F_2,v_2} \circ \mathbf{\Theta}_{F_1,v_1} = \mathbf{\Theta}_{F_1 \oplus (F_2 \odot v_1), v_2 \odot v_1} \quad .$$

La figure 2.11 présente deux transformations de $\mathbb{Z}^2$ et leurs compositions.

On note $(\Theta, \circ)$ le monoïde des plongements de $\mathbb{Z}^d$ dans lui-même munis de la composition. Un élément $\mathbf{\Theta}_{F,v}$ de ce monoïde est inversible si et seulement si $|F| = 1$.

La transformation géométrique sympathique de découpage spatial associée à $\mathbf{\Theta}_{F,v}$ est notée $\mathsf{P}_{F,v}$ et vérifie pour tout temps t et toute position p :



$$\mathsf{P}_{F,v}(t,p) \quad = \quad t \circledast (F \oplus (p \odot v)) \quad .$$

Soit $\mathcal{A}$ un automate cellulaire. On note $o_{F,v}$ l'application bijective de $\left(Q^{|F|}\right)^{\mathbb{Z}^d}$ dans $Q^{\mathbb{Z}^d}$ d'ouverture de la transformation $\mathbf{\Theta}_{F,v}$, c'est-à-dire l'application dont l'inverse vérifie,

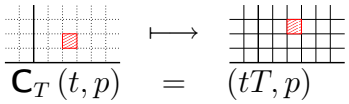
pour toute configuration C de $\mathcal{A}$ et pour toute position $p \in \mathbb{Z}^d$, l'équation

$$o_{F,v}^{-1}(C)_p = \left(C_{\mathbf{\Theta}_{F,v}(p)_1}, \ldots, C_{\mathbf{\Theta}_{F,v}(p)_{|F|}}\right) \quad .$$

Le transformé de l'automate cellulaire $\mathcal{A}$ par le découpage spatial $\mathsf{P}_{F,v}$ est l'automate cellulaire de règle globale de transition

$$o_{F,v}^{-1} \circ G_{\mathcal{A}} \circ o_{F,v} \quad .$$

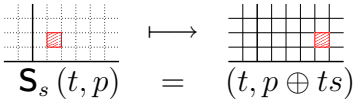
Découpage temporel. Les transformations géométriques sympathiques de découpage temporel sont notées C_T, où $T \in \mathbb{N}^*$, et vérifient, pour tout temps t et toute position p :



Le transformé de l'automate cellulaire $\mathcal{A}$ par le découpage temporel C_T est l'automate cellulaire de règle globale de transition

$$G_{\mathcal{A}}^T \quad .$$

Translation. Les transformations géométriques sympathiques de translation sont notées S_s, où $s \in \mathbb{Z}^d$, et vérifient, pour tout temps t et toute position p :



Le transformé de l'automate cellulaire $\mathcal{A}$ par le découpage temporel S_s est l'automate cellulaire de règle globale de transition

$$\sigma_s \circ G_{\mathcal{A}} \quad .$$

Composition. On introduit les transformations PCS en posant

$$\mathsf{PCS}_{F,v,T,s} = \mathsf{P}_{F,v} \circ \mathsf{S}_s \circ \mathsf{C}_T \quad .$$

Pour tout temps t et toute position p, ces transformations vérifient

$$\mathsf{PCS}_{F,v,T,s}(t,p) = tT \circledast (F \oplus (p \odot v \oplus ts)) \quad .$$

- On rappelle que les transformations géométriques sont définies du diagramme espace-temps d'arrivée dans le diagramme espace-temps de départ. Aussi, l'ordre de la composition semble inversé.

Le transformé de l'automate cellulaire $\mathcal{A}$ par la transformation $\mathsf{PCS}_{F,v,T,s}$ est l'automate cellulaire de règle globale de transition

$$o_{F,v}^{-1} \circ \sigma_s \circ G_{\mathcal{A}}^T \circ o_{F,v} \quad .$$

L'ensemble des transformations PCS est engendré par les transformations P, C et S et est stable par composition. En effet, on vérifie par le calcul que

$$\mathsf{PCS}_{F_2,v_2,T_2,s_2} \circ \mathsf{PCS}_{F_1,v_1,T_1,s_1} = \mathsf{PCS}_{F_1 \oplus (F_2 \odot v_1), v_2 \odot v_1, T_1 T_2, (s_2 \odot v_1) \oplus T_2 s_1} \quad .$$

Caractérisation des transformations sympathiques

Les transformations définies ci-dessus semblent très particulières. En fait, à cause de la nature très contrainte des automates cellulaires, elles correspondent exactement aux transformations sympathiques.

Théorème 13. *Une transformation géométrique est sympathique si et seulement si elle s'exprime comme une transformation* **PCS**.

Démonstration. Pour montrer que les transformations **PCS** sont des transformations sympathiques, la définition impose qu'elles satisfassent deux critères. La première condition, l'existence d'un unique automate cellulaire transformé complètement défini est assurée par l'exhibition explicite de l'automate cellulaire transformé, donnée au paragraphe précédent. La deuxième condition, plus technique, est aisément vérifiée car pour toute transformation **PCS** $\mathbf{\Lambda}$ et tout temps t on a $\widetilde{\mathbf{\Lambda}}\left(\{t\}\times\mathbb{Z}^d\right)\cap\widetilde{\mathbf{\Lambda}}\left(\{t+1\}\times\mathbb{Z}^d\right)=\emptyset$.

Montrons maintenant que toute transformation géométrique sympathique est une transformation **PCS**.

Les transformations géométriques sympathiques $(k,\mathbf{\Lambda})$ vérifient deux conditions :

1. pour tout automate cellulaire $\mathcal{A}$, il existe un automate cellulaire $\mathcal{B}$ vérifiant
$$\left\{\overline{\mathbf{\Lambda}}_{Q_{\mathcal{A}}}(\Delta)\right\}_{\Delta\in\mathrm{Diag}(\mathcal{A})}=\mathrm{Diag}(\mathcal{B})$$
2. pour tout temps $t\in\mathbb{N}$, $\widetilde{\mathbf{\Lambda}}(\{t+1\}\times\mathbb{Z}^d)\not\subseteq\widetilde{\mathbf{\Lambda}}(\{t\}\times\mathbb{Z}^d)$.

La technique de démonstration consiste à exhiber une série de contre-exemples, qui ne vérifient pas la première condition, pour montrer des propriétés par l'absurde.

Soit $(k,\mathbf{\Lambda})$ une transformation géométrique sympathique.

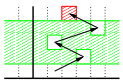
(1) On commence par montrer la propriété suivante :

$$\forall t>0,\forall(t',p_{t'})\in\widetilde{\mathbf{\Lambda}}\left(\{t\}\times\mathbb{Z}^d\right)\setminus\widetilde{\mathbf{\Lambda}}\left(\{t-1\}\times\mathbb{Z}^d\right),$$
$$\exists t''<t',\quad\{t''\}\times\mathbb{Z}^d\subseteq\widetilde{\mathbf{\Lambda}}\left(\{t-1\}\times\mathbb{Z}^d\right)$$

Cette propriété affirme que les transformations sympathiques respectent certaines dépendances temporelles. Dans un diagramme espace-temps transformé, la valeur de cellules au temps t est déterminée par de la valeur des cellules au temps $t-1$.

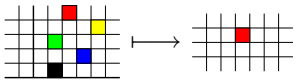
Supposons que cette propriété ne soit pas vérifiée au temps t. Alors, il existe une suite de positions spatiales $(p_0,\ldots,p_{t'})$ telles que, pour tout indice i, la cellule (i,p_i) n'est utilisée par aucune cellule du diagramme espace-temps transformé au temps $t-1$ et la cellule $(t',p_{t'})$ est utilisée par une cellule du diagramme espace-temps transformé au temps t.

$$\begin{cases}\forall i\leqslant t', & (i,p_i)\notin\widetilde{\mathbf{\Lambda}}\left(\{t-1\}\times\mathbb{Z}^d\right)\\ & (t',p_{t'})\in\widetilde{\mathbf{\Lambda}}\left(\{t\}\times\mathbb{Z}^d\right)\end{cases}$$



Considérons alors un automate cellulaire $\mathcal{A}$ d'ensemble d'états $\{\perp,0,\ldots,t'\}$, de rayon de voisinage au moins $\max_{i,j}|p_j-p_i|$ et tel que, partant d'une configuration initiale remplie de l'état $\perp$ en toute position sauf en la position p_0 contenant l'état 0 il produise un

diagramme espace-temps Δ dont les t' premières configurations sont remplies de l'état $\perp$ sauf pour tout temps i en la position p_i où il contient l'état i.



Le diagramme espace-temps transformé $\overline{\mathbf{\Lambda}}_{S_{\mathcal{A}}}(\Delta)$ n'est pas le diagramme espace-temps d'un automate cellulaire car la configuration au temps $t-1$ est monochromatique et la configuration au temps t ne l'est pas. Un automate cellulaire ne peut pas briser une telle symétrie.

(2) On montre ensuite une propriété des configurations initiales :

$$\widetilde{\mathbf{\Lambda}}\left(\{0\}\times\mathbb{Z}^d\right)=\{0\}\times\mathbb{Z}^d$$

Cette propriété affirme que la configuration initiale d'un diagramme espace-temps transformé est obtenue par transformation de la configuration initiale du diagramme espace-temps original.

C'est une simple conséquence des hypothèses sur les transformations sympathiques. En transformant tous les diagrammes espace-temps d'un automate cellulaire $\mathcal{A}$, on doit obtenir toutes les configurations initales possibles de l'automate cellulaire transformé $\mathcal{B}$, c'est-à-dire $\left(Q_{\mathcal{A}}^k\right)^{\mathbb{Z}^d}$.

Supposons que la propriété ne soit pas vérifiée. Soit $(0,p)$ une cellule initiale telle qu'il existe $(t,p')\in\widetilde{\mathbf{\Lambda}}(0,p)$ une cellule de son image avec t différent de 0. Soit $\mathcal{A}$ l'automate cellulaire à deux états $\{0,1\}$ tel que $\delta_{\mathcal{A}}$ est constant égal à 0. Pour tout diagramme espace-temps de $\mathcal{A}$ la cellule (t,p') a pour état 0, ce qui contredit le point 1 de la définition 3.

Par un raisonnement similaire, on montre que les images des cellules au temps 0 sont composées de blocs disjoints de cellules :

$$\forall p,q,\quad \operatorname{card}\left(\widetilde{\mathbf{\Lambda}}(0,p)\right)=k\quad\wedge\quad p\neq q\Rightarrow\widetilde{\mathbf{\Lambda}}(0,p)\cap\widetilde{\mathbf{\Lambda}}(0,q)=\emptyset$$

(3) On étend la propriété précédente à tous les instants :

$$\forall t\in\mathbb{N},\exists t'\in\mathbb{N},\quad\widetilde{\mathbf{\Lambda}}\left(\{t\}\times\mathbb{Z}^d\right)=\{t'\}\times\mathbb{Z}^d$$

Soit $t\in\mathbb{N}$ un instant et soit, par **(1)**, un temps t' tel que

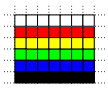
$$\widetilde{\mathbf{\Lambda}}\left(\{t\}\times\mathbb{Z}^d\right)\supseteq\{t'\}\times\mathbb{Z}^d.$$

On commence par montrer que toute cellule au temps t participe pour un même nombre l de cellules aux cellules images situées au temps t'. Formellement :

$$\begin{aligned}&\exists l,\forall p,\exists i_1,\ldots,i_l,\quad(\mathbf{\Lambda}(t,p)_{i_1},\ldots,\mathbf{\Lambda}(t,p)_{i_l})\in\left(\{t'\}\times\mathbb{Z}^d\right)^l\\ \wedge\quad&\forall i\notin\{i_1,\ldots,i_l\},\quad\mathbf{\Lambda}(t,p)_i\notin\{t'\}\times\mathbb{Z}^d\end{aligned}$$

Considérons l'automate cellulaire autarcique $\mathcal{A}$ d'ensemble d'états $\{0, \ldots, t'+1\}$ dont la règle locale de transition δ vérifie :

$$\forall i, \quad \delta(i) = \begin{cases} i+1 & \text{if } i < t'+1 \\ t'+1 & \text{if } i = t'+1 \end{cases}$$



Soit Δ le diagramme espace-temps de $\mathcal{A}$ engendré par la configuration monochromatique de couleur 0. De **(2)**, on déduit que la configuration initiale de $\Delta' = \overline{\mathbf{\Lambda}}_{Q_{\mathcal{A}}}(\Delta)$ est elle aussi monochromatique et que donc, par des considérations de symétrie, les configurations de Δ' sont monochromatiques à tout instant. En particulier, toutes les cellules de la configuration au temps t contiennent le même nombre l de composantes dans l'état t'. Il reste à montrer que nécessairement $l = k$.

Supposons que $l < k$. Soit $\mathcal{A}$ l'automate cellulaire identité d'ensemble d'états $\{0, 1\}$. Soit $\mathcal{B}$ le transformé de $\mathcal{A}$ par $(k, \mathbf{\Lambda})$. Soit Δ'_n le diagramme espace-temps de $\mathcal{B}$ dont la configuration initiale est remplie par l'état $(0, \ldots, 0)$ sauf en une boule de rayon n centrée en 0 qui est, elle, remplie de l'état $(1, \ldots, 1)$. Soit Δ_n le diagramme espace-temps de $\mathcal{A}$ dont le transformé est Δ'_n. De **(2)**, on déduit que la configuration initiale de Δ_n contient exactement kn^d cellules dans l'état 1. La configuration Δ_n au temps t' est identique à la configuration initiale. De **(1)** on déduit que la configuration de Δ'_n au temps t contient au moins $\lceil \frac{k}{l} n^d \rceil$ cellules dans un état distinct de $(0, \ldots, 0)$. Ce qui implique que le rayon de voisinage de $\mathcal{B}$ est au moins

$$r_n \geqslant \frac{\lceil \frac{k}{l} n^d \rceil^{1/d} - n}{t}$$

Comme $\frac{k}{l} > 1$, la suite (r_n) est non bornée aussi $\mathcal{B}$ ne peut pas être un automate cellulaire.

(4) On étend de même la propriété des blocs disjoints :

$$\forall t, \forall p, q, \quad \text{card}\left(\widetilde{\mathbf{\Lambda}}(t,p)\right) = k \quad \wedge \quad p \neq q \Rightarrow \widetilde{\mathbf{\Lambda}}(t,p) \cap \widetilde{\mathbf{\Lambda}}(t,q) = \emptyset$$

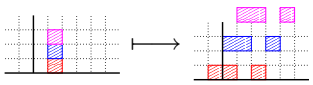
On montre d'abord que $\text{card}\left(\widetilde{\mathbf{\Lambda}}(t,p)\right) = k$, c'est-à-dire que les composantes d'une même cellule image correspondent à des cellules disjointes. Soit $\mathcal{A}$ l'automate cellulaire identité d'ensemble d'états $\{1, \ldots, k\}$. Soit Δ un diagramme espace-temps de $\mathcal{A}$ tel que son transformé Δ' possède une configuration initale monochromatique remplie de l'état $(1, \ldots, k)$. Par symétrie, la configuration de Δ' au temps t est monochromatique remplie par un état q. De **(1)** on déduit que q doit contenir toutes les composantes possibles 1, 2, …, k et donc que q ne peut être qu'une permutation de $(1, \ldots, k)$. Chaque composante correspond donc à des cellules distinctes.

On montre maintenant la propriété suivante :

$$\forall n, \forall t, \forall p, q, \quad (p \neq q \wedge |p - q| \leqslant n) \Rightarrow \widetilde{\mathbf{\Lambda}}(t,p) \cap \widetilde{\mathbf{\Lambda}}(t,q) = \emptyset$$

On peut l'obtenir par simple généralisation des considérations de symétrie précédentes. Soit n un entier fixé. Soit $\mathcal{A}_n$ l'automate cellulaire identité d'ensemble d'états $\{1, \ldots, n^d k\}$. Soit Δ_n un diagramme espace-temps de $\mathcal{A}_n$ dont le transformé Δ'_n possède une configuration initiale périodique dont le motif périodique est la boule de dimension d de rayon n remplie par les états $(1, \ldots, k)$ à $((n^d - 1)k + 1, \ldots, n^d k)$. Par symétrie, la configuration de Δ'_n au temps t est aussi périodique et sa période est plus petite. De **(1)**, on déduit que le motif périodique de cette configuration contient tous les états possibles 1, 2, …, $n^d k$, donc toutes les cellules à distance inférieure à n correspondent à des cellules disjointes.

(5) Ensuite, on montre une propriété d'uniformité dans le temps :



$$\forall p, \exists s_0, T_0, \forall t, \quad \mathbf{\Lambda}(t,p) = tT_0 \circledast (\pi_2(\mathbf{\Lambda}(0,p)) \oplus ts_0)$$

Cette propriété affirme simplement que les images de cellules de même position sont arrangées régulièrement au cours du temps.

Soit p une position de l'espace. On introduit $(p_1, \ldots, p_k)$, $(p'_1, \ldots, p'_k)$ et t tels que $\mathbf{\Lambda}(0,p) = ((0,p_1), \ldots, (0,p_k))$ et $\mathbf{\Lambda}(1,p) = ((t,p'_1), \ldots, (t,p'_k))$.

Considérons l'automate cellulaire $\mathcal{A}$ d'ensemble d'états $\{\perp, 1, \ldots, k, W_1, \ldots, W_{t-1}\}$, de rayon de voisinage $\max\{2\|p_1\|_\infty, \ldots, 2\|p_k\|_\infty, 2\|p'_1\|_\infty, \ldots, 2\|p'_k\|_\infty\}$ et dont un des diagrammes espace-temps Δ est tel que pour tout temps de 0 à t le diagramme espace-temps soit rempli par l'état $\perp$ sauf :

$$\begin{aligned} \Delta(0,p_1) &= 1, & \ldots, \quad \Delta(0,p_k) &= k, \\ \Delta(1,0) &= W_1, & \ldots, \quad \Delta(t-1,0) &= W_{t-1}, \\ \Delta(t,p'_1) &= 1, & \ldots, \quad \Delta(t,p'_k) &= k. \end{aligned}$$

Toutes les transitions non définies donnent l'état $\perp$. Soit Δ' le diagramme espace-temps transformé de Δ. Par construction, aux temps 0 et 1, les configurations de Δ' sont remplies par l'état $(\perp, \ldots, \perp)$ sauf en position p qui contiennent l'état $(1, \ldots, k)$. Comme Δ' est le diagramme espace-temps d'un automate cellulaire, ses configurations aux temps 0 et 1 étant les mêmes, elles sont les mêmes à tous les temps. Par construction, cela implique que toutes les cellules ont la même forme : $\exists s, \quad p_1 = p'_1 \oplus s \quad \wedge \quad \ldots \quad \wedge \quad p_k = p'_k \oplus s$.

(6) On montre que la même proposition est vraie à une translation près :

$$\forall i, p, \exists s_i, T_i, \forall t, \quad \mathbf{\Lambda}(t, p \oplus t\varsigma_i) = tT_i \circledast (\pi_2(\mathbf{\Lambda}(0,p)) \oplus ts_i)$$

La démonstration est exactement la même que pour la proposition précédente en considérant des cellules translatées au temps 1.

(7) Enfin, on conclut :

$$\mathbf{\Lambda} = \mathsf{PCS}_{\pi_2(\mathbf{\Lambda}(0,0)),(s_1 \ominus s_0, \ldots, s_d \ominus s_0), T_0, s_0}$$

La démonstration de cette dernière propriété repose simplement sur le fait que les translations forment une base de déplacement dans $\mathbb{Z}^d$. ■

2.3.2 Axiomatisation

Première tentative

On propose de définir un modèle du groupage abstrait reposant sur les transformations **PCS** et la relation sous-automate de la manière suivante :

Obj est l'ensemble des automates cellulaires de dimension d ;

Trans est l'ensemble des transformations **PCS** de dimension d ;

apply est l'opérateur de transformation des automates cellulaires ;

divide est la relation sous-automate ;

combine est la composition des transformations **PCS** (*i.e.* **combine**$(\mathcal{T}_1, \mathcal{T}_2) = \mathcal{T}_2 \circ \mathcal{T}_1$).

Il faut à présent regarder si les axiomes sont vérifiés :

Combinaison. La structure (Trans, $\cdot$) est un monoïde. C'est une propriété de la composition et de la stabilité des transformations **PCS** par composition.

Compatibilité. L'opérateur **apply** fait agir le monoïde (Trans, $\cdot$) sur l'ensemble Obj. La composition des transformations a été définie par l'intermédiaire de la composition de leurs applications.

Divisibilité. La relation **divide** est une relation de préordre. On sait que la relation sous-automate est une relation de préordre.

Transitivité. L'opérateur **apply** est compatible avec la relation **divide**. Soient $\mathcal{A}$ et $\mathcal{B}$ deux automates cellulaires tels que $\mathcal{A} \mid \mathcal{B}$. Soit φ une application injective de $Q_{\mathcal{A}}$ dans $Q_{\mathcal{B}}$ telle que $\overline{\varphi} \circ G_{\mathcal{A}} = G_{\mathcal{B}} \circ \overline{\varphi}$. Soit $\mathcal{T} = \mathbf{PCS}_{F,v,T,s}$ une transformation **PCS**. Pour montrer que $\mathcal{A}^{\mathcal{T}} \mid \mathcal{B}^{\mathcal{T}}$, exhibons une application injective ψ de $Q_{\mathcal{A}}^{|F|}$ dans $Q_{\mathcal{B}}^{|F|}$ telle que $\overline{\psi} \circ G_{\mathcal{A}}^{\mathcal{T}} = G_{\mathcal{B}}^{\mathcal{T}} \circ \overline{\psi}$. Une solution consiste à prendre pour ψ le produit canonique de φ sur $Q_{\mathcal{A}}^{|F|}$.

Surjectivité. L'opérateur **apply** préserve la richesse des objets. Soit $\mathcal{A}$ un automate cellulaire de voisinage v de taille ν. Soit $\mathcal{T} = \mathbf{PCS}_{F,v',T,s}$ une transformation **PCS**. On définit l'automate cellulaire $\mathcal{B}$ à partir de l'automate cellulaire $\mathcal{A}$ de la manière suivante. Le voisinage de $\mathcal{B}$ est $v \odot v' \ominus s$ auquel on ajoute 0 en premier vecteur, son ensemble d'états est $(Q_{\mathcal{A}} \cup \{\perp\}) \times \mathbb{Z}_T$ et sa règle locale de transition δ' est définie (partiellement) pour tout $(\nu+1)$-uple $(x_0, x_1, \ldots, x_\nu)$ d'éléments de $Q_{\mathcal{A}} \cup \{\perp\}$ et tout entier $i \in \mathbb{Z}_T$ de la manière suivante :

$$\delta'\left((x_0,i),(x_1,i),\ldots,(x_\nu,i)\right) = \begin{cases} (\delta_{\mathcal{A}}(x_1,\ldots,x_\nu),0) & \text{si } i = T-1 \\ & \text{et } \perp \notin \{x_0,\ldots,x_\nu\}\,; \\ (x_0, i+1) & \text{sinon.} \end{cases}$$

Ainsi $\mathcal{A}$ est un sous-automate de $\mathcal{B}^{\mathcal{T}}$ par l'application injective φ de $Q_{\mathcal{A}}$ dans $Q_{\mathcal{B}}^{|F|}$ définie pour tout état q de $Q_{\mathcal{A}}$ par

$$\varphi(q) = ((q,0),(\perp,0),\ldots,(\perp,0)) \quad .$$

Proximité. L'opérateur **apply** n'éloigne pas trop les transformations d'un même objet. Cet axiome n'est pas vérifié ! En effet, il est possible de trouver des couples de transformations spatiales $\mathbf{\Theta}_{F_1,v_1}$, $\mathbf{\Theta}_{F_2,v_2}$ tels que

$$(\Theta \circ \mathbf{\Theta}_{F_1,v_1}) \cap (\Theta \circ \mathbf{\Theta}_{F_2,v_2}) = \emptyset \quad .$$

▶ Pour s'en convaincre, voici un exemple en dimension 1. On pose $F_1 = (0,1)$, $v_1 = 2$, $F_2 = (0,3)$, $v_2 = 2$ (ces transformations sont représentées sur la figure 2.12). Pour toute transformation spatiale $\mathbf{\Theta}_{F'_1,v'_1} \in (\Theta \circ \mathbf{\Theta}_{F_1,v_1})$, si p est la plus petite position apparaissant dans le uple F'_1, alors $p+1$ apparaît aussi dans F'_1. Pour toute transformation spatiale $\mathbf{\Theta}_{F'_2,v'_2} \in (\Theta \circ \mathbf{\Theta}_{F_2,v_2})$, si p est la plus petite position apparaissant dans le uple F'_2, alors $p+1$ n'apparaît pas dans F'_2.

FIG. 2.12 – Contre-exemple pour l'axiome de **Proximité**

Ce problème est lié à la trop grande liberté qui est donnée aux formes des motifs des transformations spatiales. Une solution consiste à se restreindre à un sous-monoïde de $(\Theta, \circ)$ qui ne possède pas ce défaut et qui engendre des transformations géométriques qui vérifient toujours les précédents axiomes.

On choisit[2] de se restreindre aux plongements de la forme $\mathbf{\Theta}_{\prod_{i=1}^{d} [\![0,m_i-1]\!], v \otimes (m_1,\ldots,m_d)}$ où les m_i sont des entiers strictement positifs et les vecteurs v_i de la famille libre de vecteurs v ont toutes leurs coordonnées égales à 0 sauf une égale à 1. Si on note ς_i le vecteur de $\mathbb{Z}^d$ dont toutes les coordonnées sont égales à 0 sauf la i-ème qui est égale à 1, on peut toujours trouver une permutation τ de $[\![1,d]\!]$ telle que pour tout indice i, on ait $v_i = \varsigma_{\tau(i)}$. Ces plongements correspondent à des découpages de la grille par homothétie puis permutation éventuelle des vecteurs de base. En dimension 1, cela correspond au découpage de segments de cellules du groupage carré.

On note $\tilde{\mathsf{P}}_{(m_1,\ldots,m_d),\tau}$ le découpage spatial particulier vérifiant

$$\tilde{\mathsf{P}}_{(m_1,\ldots,m_d),\tau} = \mathsf{P}_{\prod_{i=1}^{d} [\![0,m_i-1]\!], \left(\varsigma_{\tau(1)},\ldots,\varsigma_{\tau(d)}\right) \otimes m} \quad .$$

On introduit alors les transformations $\tilde{\mathsf{P}}\mathsf{CS}$ en posant

$$\tilde{\mathsf{P}}\mathsf{CS}_{m,\tau,T,s} = \tilde{\mathsf{P}}_{m,\tau} \circ \mathsf{S}_s \circ \mathsf{C}_T \quad .$$

L'ensemble des transformations $\tilde{\mathsf{P}}\mathsf{CS}$ est stable par composition, ainsi

$$\tilde{\mathsf{P}}\mathsf{CS}_{(m'_1,\ldots,m'_d),\tau',T',s'} \circ \tilde{\mathsf{P}}\mathsf{CS}_{(m_1,\ldots,m_d),\tau,T,s} = \tilde{\mathsf{P}}\mathsf{CS}_{(m''_1,\ldots,m''_d),\tau'',T'',s''}$$

avec les paramètres

$$\begin{aligned} m''_i &= m_i m'_{\tau(i)} \\ \tau'' &= \tau' \circ \tau \\ T'' &= TT' \\ s'' &= \left(s' \odot \left(\varsigma_{\tau(1)}, \ldots, \varsigma_{\tau(d)}\right) \otimes m\right) \oplus T's \end{aligned}$$

Hélas, les transformations $\tilde{\mathsf{P}}\mathsf{CS}$ ne satisfont pas, elles non plus, l'axiome de **Proximité** ! La cause est maintenant les translations. Le manque de corrélation entre la composante S et la composante $\tilde{\mathsf{P}}$ engendre des pathologies.

[2]Si ce choix n'en est pas réellement un en dimension 1 (on ignore simplement la transformation «image miroir»), il s'agit d'une réelle restriction dans le cas des dimensions supérieures. Nous nous restreignons au plus grand sous-monoïdes que nous connaissions, qui contienne les transformations présentées en début de chapitre, et qui puisse satisfaire l'axiome de **Proximité**.

- Pour s'en convaincre, plaçons-nous en dimension 1 et notons $\langle m, n, k\rangle$ la transformation $\tilde{\mathsf{PCS}}_{m,\mathrm{id},n,k}$. Soient $\langle m, n, k\rangle$, $\langle m', n', k'\rangle$ et $\langle m'', n'', k''\rangle$, $\langle m''', n''', k'''\rangle$ quatre transformations :

$$\begin{aligned}\langle m'', n'', k''\rangle \circ \langle m, n, k\rangle &= \langle mm'', nn'', mk'' + kn''\rangle \\ \langle m''', n''', k'''\rangle \circ \langle m', n', k'\rangle &= \langle m'm''', n'n''', m'k''' + k'n'''\rangle\end{aligned}$$

  Pour essayer de vérifier l'axiome de proximité, on pose $m'' = m'$, $m''' = m$, $n'' = \alpha n'$ et $n''' = \alpha n$ pour un entier α. Les équations deviennent alors :

$$\begin{aligned}\langle m'', n'', k''\rangle \circ \langle m, n, k\rangle &= \langle mm', \alpha nn', mk'' + \alpha kn'\rangle \\ \langle m''', n''', k'''\rangle \circ \langle m', n', k'\rangle &= \langle mm', \alpha nn', m'k''' + \alpha k'n\rangle\end{aligned}$$

  On cherche k'', k''' et α tels que $mk'' - m'k''' = \alpha(kn' - k'n)$, ce qui est possible, mais l'axiome de **Proximité** ajoute la contrainte que k'' et α ne dépendent que de $\langle m', n', k'\rangle$. Pour que cela soit possible, il faudrait que k et n (respectivement k' et n') soient proportionnels à m (respectivement m').

Tout n'est cependant pas perdu, nous allons utiliser les transformations $\tilde{\mathsf{PCS}}$. Observons ce qui se passe lorsqu'on compose une transformation C et une transformation $\tilde{\mathsf{PCS}}$:

$$\mathsf{C}_{T'} \circ \tilde{\mathsf{PCS}}_{m,\tau,T,s} = \tilde{\mathsf{PCS}}_{m,\tau,T'T,T's} \quad .$$

En choisissant T' de façon astucieuse, par exemple en prenant $\prod_i m_i$, on peut corréler la translation et le découpage temporel avec le découpage spatial. Aussi, on se convainc aisément que si on se restreint à un sous-monoïde de transformations raisonnablement corrélées de $\tilde{\mathsf{PCS}}$, la relation de comparaison engendrée est exactement la même que pour les transformations $\tilde{\mathsf{PCS}}$. On peut donc travailler avec les transformations $\tilde{\mathsf{PCS}}$ si on trouve des transformations corrélées qui satisfassent les axiomes. On choisit les transformations $\tilde{\mathsf{PCS}}$' définies ainsi :

$$\tilde{\mathsf{PCS}}'_{m,\tau,T,s} = \tilde{\mathsf{PCS}}_{m,\tau,(\prod_i m_i)T,s\odot(\varsigma_{\tau(1)},\ldots,\varsigma_{\tau(d)})\otimes m}$$

L'ensemble des transformations $\tilde{\mathsf{PCS}}$' est stable par composition car

$$\tilde{\mathsf{PCS}}'_{(m'_1,\ldots,m'_d),\tau',T',s'} \circ \tilde{\mathsf{PCS}}'_{(m_1,\ldots,m_d),\tau,T,s} = \tilde{\mathsf{PCS}}'_{(m''_1,\ldots,m''_d),\tau'',T'',s''}$$

avec les paramètres

$$\begin{aligned}m''_i &= m_i m'_{\tau(i)} \\ \tau'' &= \tau' \circ \tau \\ T'' &= TT'\end{aligned}$$

et il faut décomposer

$$\begin{aligned}&\left(s' \odot (\varsigma_{\tau'(1)}, \ldots, \varsigma_{\tau'(d)}) \otimes m' \odot \left(\varsigma_{\tau(1)}, \ldots, \varsigma_{\tau(d)}\right) \otimes m\right) \\ \oplus&(\prod_i m'_i)T' \left(s \odot (\varsigma_{\tau(1)}, \ldots, \varsigma_{\tau(d)}) \otimes m\right)\end{aligned}$$

sous la forme

$$s'' \odot (\varsigma_{\tau''(1)}, \ldots, \varsigma_{\tau''(d)}) \otimes m'' \quad .$$

On traite les deux parties de l'expression séparément. D'une part,

$$(\varsigma_{\tau''(1)}, \ldots, \varsigma_{\tau''(d)}) \otimes m'' = (\varsigma_{\tau'(1)}, \ldots, \varsigma_{\tau'(d)}) \otimes m' \odot \big(\varsigma_{\tau(1)}, \ldots, \varsigma_{\tau(d)}\big) \otimes m$$

et d'autre part,

$$\left((\prod_i m'_i) s \right) \odot (\varsigma_{\tau(1)}, \ldots, \varsigma_{\tau(d)}) \otimes m$$

se décompose sans difficulté en un terme du type

$$\bar{s} \odot (\varsigma_{\tau''(1)}, \ldots, \varsigma_{\tau''(d)}) \otimes m'' \quad .$$

Pour cela, il suffit de poser, pour tout indice i,

$$\bar{s}_i = \frac{(\prod_i m'_i)}{m_{\tau'(i)}} s_{\tau'(i)}.$$

Finalement, on peut réécrire s'' sous la forme :

$$s'' = s' \oplus T'\bar{s} \quad .$$

Le point important ici est de remarquer que s'' s'exprime comme une somme de s' et d'un autre terme. Aussi, si on est libre du choix de s' et que le deuxième terme est connu, on peut faire prendre n'importe quelle valeur à s''.

Deuxième tentative

On propose de définir un modèle du groupage abstrait reposant sur les transformations $\tilde{\mathsf{P}}\mathsf{CS}$ et la relation sous-automate de la manière suivante :

Obj est l'ensemble des automates cellulaires de dimension d ;

Trans est l'ensemble des transformations $\tilde{\mathsf{P}}\mathsf{CS'}$ de dimension d ;

apply est l'opérateur de transformation des automates cellulaires ;

divide est la relation sous-automate ;

combine est la composition des transformations $\tilde{\mathsf{P}}\mathsf{CS'}$ (*i.e.* $\mathsf{combine}(\mathcal{T}_1, \mathcal{T}_2) = \mathcal{T}_2 \circ \mathcal{T}_1$).

Tous les axiomes vérifiés plus haut dans le cas des transformations **PCS** restent vérifiés si on se restreint aux transformations $\tilde{\mathsf{P}}\mathsf{CS'}$. Il ne reste donc à vérifier que l'axiome de **Proximité**.

On montre que l'opérateur **apply** n'éloigne pas trop les transformations d'un même objet. Soit $\mathcal{A}$ un automate cellulaire et $\tilde{\mathsf{P}}\mathsf{CS'}_{m,\tau,T,s}$ et $\tilde{\mathsf{P}}\mathsf{CS'}_{m',\tau',T',s'}$ deux transformations $\tilde{\mathsf{P}}\mathsf{CS'}$. On se propose d'exhiber deux transformations $\tilde{\mathsf{P}}\mathsf{CS'}_{m'',\tau'',T'',s''}$ et $\tilde{\mathsf{P}}\mathsf{CS'}_{m''',\tau''',T''',s'''}$ telles que la première ne dépend que de $\tilde{\mathsf{P}}\mathsf{CS'}_{m',\tau',T',s'}$ et qu'elles vérifient

$$\tilde{\mathsf{P}}\mathsf{CS'}_{m'',\tau'',T'',s''} \circ \tilde{\mathsf{P}}\mathsf{CS'}_{m,\tau,T,s} = \tilde{\mathsf{P}}\mathsf{CS'}_{m''',\tau''',T''',s'''} \circ \tilde{\mathsf{P}}\mathsf{CS'}_{m',\tau',T',s'} \quad .$$

Pour cela, on pose :

$$\begin{aligned} m''_i &= \prod_{k=1}^{d} m'_k \\ \tau'' &= \mathrm{id} \\ T'' &= T' \\ s'' &= 0 \end{aligned}$$

$$\begin{aligned} m_i''' &= \frac{m_{\tau'^{-1}(i)} \prod_{k=1}^{d} m'_k}{m'_{\tau'^{-1}(i)}} \\ \tau''' &= \tau \circ \tau'^{-1} \\ T''' &= T \end{aligned}$$

Et on se convainc, grâce à la discussion sur les translations et la composition de transformations $\tilde{\mathsf{P}}\mathsf{CS'}$ à la fin du paragraphe précédent, qu'il existe une valeur de s''' qui satisfait l'équation.

2.4 Une extension du groupage carré

2.4.1 Définitions

Nous définissons une extension du groupage à partir des transformations géométriques $\tilde{\mathsf{P}}\mathsf{CS}$, une sous-famille des transformations PCS qui induit une relation de préordre.

Par analogie avec les transformations présentées en début de chapitre (il y a correspondance dans le cas de la dimension 1), on notera par la suite $\langle m_\tau, n, k\rangle$ la transformation $\tilde{\mathsf{P}}\mathsf{CS}_{m,\tau,n,k}$, séparant ainsi les composantes spatiale, temporelle et de translation. De même $\mathcal{A}^{\langle m_\tau,n,k\rangle}$ dénotera l'automate cellulaire transformé de l'automate cellulaire $\mathcal{A}$ par cette transformation, de règle globale de transition

$$G_{\mathcal{A}}^{\langle m_\tau,n,k\rangle} = o_{m_\tau}^{-1} \circ \sigma_k \circ G_{\mathcal{A}}^{n} \circ o_{m_\tau}$$

où o_{m_τ} est une notation simplifiée pour $o_{\prod_{i=1}^{d}[\![0,m_i-1]\!],\left(\varsigma_{\tau(1)},\ldots,\varsigma_{\tau(d)}\right)\otimes m}$.

Définition 4. *Un automate cellulaire $\mathcal{A}$ est simulé par un automate cellulaire $\mathcal{B}$, noté $\mathcal{A} \leqslant \mathcal{B}$, s'il existe deux transformations $\tilde{\mathsf{P}}\mathsf{CS}$ $\langle m_\tau, n, k\rangle$, $\langle m'_{\tau'}, n', k'\rangle$ telles que*

$$\mathcal{A}^{\langle m_\tau,n,k\rangle} \subseteq \mathcal{B}^{\langle m'_{\tau'},n',k'\rangle} \quad .$$

Théorème 14. *La relation de simulation $\leqslant$ est égale à la relation de simulation induite par un modèle du groupage abstrait.*

Démonstration. Montrons d'abord que si on se restreint aux transformations $\tilde{\mathsf{P}}\mathsf{CS'}$ dans la définition de la relation $\leqslant$, on définit exactement la même relation. Soient $\mathcal{A}$ et $\mathcal{B}$ deux automates cellulaires et $\langle m_\tau, n, k\rangle$, $\langle m'_{\tau'}, n', k'\rangle$ deux transformations $\tilde{\mathsf{P}}\mathsf{CS}$ vérifiant $\mathcal{A}^{\langle m_\tau,n,k\rangle} \subseteq \mathcal{B}^{\langle m'_{\tau'},n',k'\rangle}$. Les transformations PCS vérifiant l'axiome de **Transitivité**, la relation suivante est vérifiée

$$\left(\mathcal{A}^{\langle m_\tau,n,k\rangle}\right)^{\left\langle 1_{\mathrm{id}},\prod_i m_i \prod_i m'_i,0\right\rangle} \subseteq \left(\mathcal{B}^{\langle m'_{\tau'},n',k'\rangle}\right)^{\left\langle 1_{\mathrm{id}},\prod_i m_i \prod_i m'_i,0\right\rangle} \quad ,$$

relation qui se simplifie en

$$\mathcal{A}^{\left\langle m_\tau,\prod_i m_i \prod_i m'_i n,\prod_i m_i \prod_i m'_i k\right\rangle} \subseteq \mathcal{B}^{\left\langle m'_{\tau'},\prod_i m_i \prod_i m'_i n',\prod_i m_i \prod_i m'_i k'\right\rangle}$$

où les deux transformations sont des transformations $\tilde{\mathsf{P}}\mathsf{CS'}$. Ce qui conclut la démonstration car on a montré au paragraphe précédent que les transformations $\tilde{\mathsf{P}}\mathsf{CS'}$ définissaient bien un modèle du groupage abstrait. ■

Corollaire 2. *La relation $\leqslant$ est une relation de préordre.*

Dans la suite, on note $\sim$ la relation d'équivalence associée à $\leqslant$ et $<$ la relation de comparaison stricte. Par abus de notation, on notera aussi $\leqslant$ la relation d'ordre induite et $<$ la relation d'ordre stricte induite. On s'intéresse à la classification induite par les classes d'équivalence ordonnées par l'ordre induit.

2.4.2 Espace des phases

En plus de se comporter agréablement des points de vue géométrique pour les diagrammes espace temps et algébrique pour la règle globale de transition, les transformations **PCS** transforment l'espace des phases des automates cellulaires de manière « régulière ».

Dans cette thèse, nous appellerons espace des phases[3] d'une règle globale de transition G (éventuellement une restriction d'une telle fonction à une famille de configurations stables par translation et par application de la règle) le graphe dirigé dont les sommets sont les configurations de G munis de $d+1$ types d'arêtes. Soient x et y deux sommets du graphe. Une arête simple $x \to y$ de x à y signifie que y est l'image de x par G. Une arête en pointillé $x \dashrightarrow_i y$ de x à y indicée par i signifie que y est l'image de x par une translation élémentaire de vecteur ς_i. La figure 2.13 présente l'espace des phases partiel d'un automate cellulaire unidirectionnel de dimension 1.

Ce graphe possède une structure très régulière. De tout sommet part exactement une arête simple. Pour tout indice $i \in [\![1, d]\!]$, de tout sommet part et arrive exactement une arête en pointillés indicée par i. On peut donc emprunter toute arête sans ambiguïté, y compris emprunter une arête en pointillés dans le sens contraire. Un chemin dans le graphe peut donc être donné simplement par la donnée de ses extrémités et de la suite des types d'arêtes empruntées. Un chemin généralisé est un chemin dans lequel on s'autorise à emprunter des arêtes en pointillés en sens contraire.

Les règles globales de transition des automates cellulaires commutant avec les translations, ce graphe rend compte de cette commutativité de la manière suivante. Pour toute paire d'arêtes $\leftrightsquigarrow_1$, $\leftrightsquigarrow_2$, éventuellement des arêtes en pointillés en sens contraire, et pour tout sommet x, il existe un unique sommet y tel que $x \leftrightsquigarrow_1 \leftrightsquigarrow_2 y$ et $x \leftrightsquigarrow_2 \leftrightsquigarrow_1 y$.

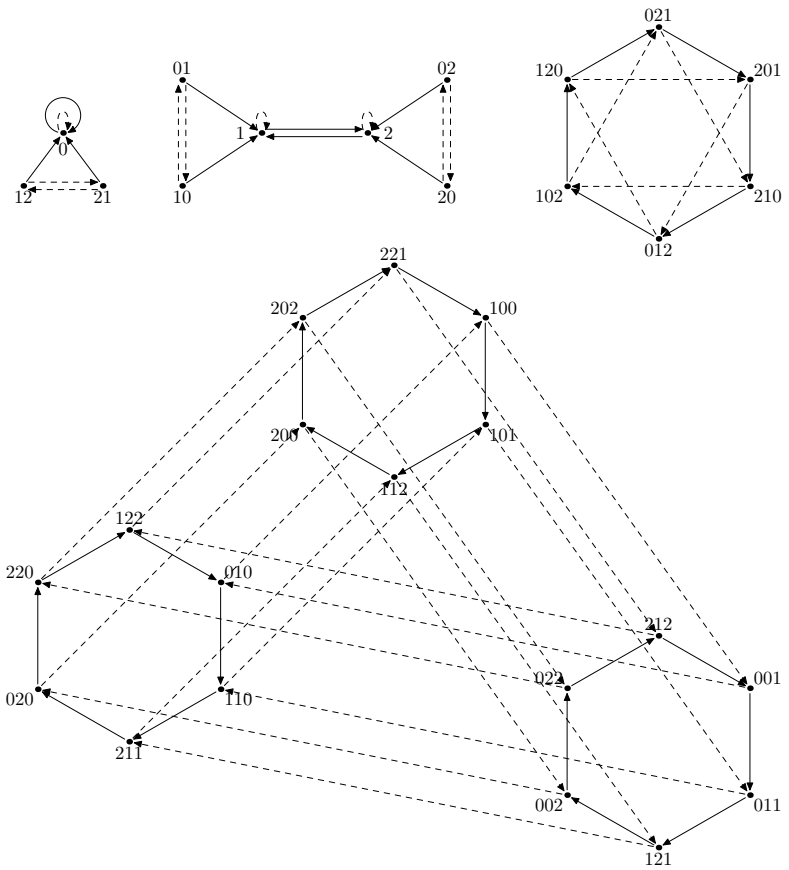
La transformation d'un automate cellulaire par une transformation **PCS** peut être vue comme une opération de réécriture sur le graphe. Voici l'effet de chaque type de transformation :

transformation temporelle C_T. Les sommets sont identiques. On définit une arête simple $x \to y$ entre deux sommets x et y s'il existe un chemin de T arêtes simples $x \underbrace{\to \cdots \to}_{T} y$ entre x et y. Les arêtes en pointillés sont conservées.

translation S_s. Les sommets sont identiques. On définit une arête simple $x \to y$ entre deux sommets x et y s'il existe un chemin généralisé de $n+m+1$ arêtes

$$x \to \underbrace{\dashrightarrow_{i_1} \dashrightarrow_{i_2} \cdots \dashrightarrow_{i_n}}_{n} \underbrace{\dashleftarrow_{i'_1} \dashleftarrow_{i'_2} \cdots \dashleftarrow_{i'_m}}_{m} y$$

[3]La notion d'espace des phases est classique dans le cadre de la théorie des systèmes dynamiques, ici nous nous intéressons à une variante spécifique aux automates cellulaires où les translations jouent un rôle particulier.



Espace des phases de l'automate cellulaire $(\mathbb{Z}, \mathbb{Z}_3, [\![0,1]\!], +)$. Seules sont représentées les configurations périodiques de période au plus 3. Les sommets du graphe sont étiquetés par le motif périodique de taille minimale apparaissant à droite de la position 0 dans la configuration correspondant au sommet.

FIG. 2.13 – Espace des phases d'un automate cellulaire

entre x et y vérifiant $s = \sum_{k=1}^{n} \varsigma_{i_k} - \sum_{k=1}^{m} \varsigma_{i'_k}$. Les arêtes en pointillés sont conservées.

transformation spatiale $\mathsf{P}_{F,v}$. Chaque sommet du graphe x est renommé en $o_{F,v}^{-1}(x)$. On définit une arête en pointillés $o_{F,v}^{-1}(x) \dashrightarrow_i o_{F,v}^{-1}(y)$ indicée par $i \in [\![1,d]\!]$ entre deux sommets $o_{F,v}^{-1}(x)$ et $o_{F,v}^{-1}(y)$ s'il existe un chemin généralisé de $n+m$ arêtes

$$x \underbrace{\dashrightarrow_{j_1} \dashrightarrow_{j_2} \cdots \dashrightarrow_{j_n}}_{n} \underbrace{\dashleftarrow_{j'_1} \dashleftarrow_{j'_2} \cdots \dashleftarrow_{j'_m}}_{m} y$$

entre x et y vérifiant $v_i = \sum_{k=1}^{n} \varsigma_{j_k} - \sum_{k=1}^{m} \varsigma_{j'_k}$. Les arêtes simples sont conservées.

2.4.3 Exploration

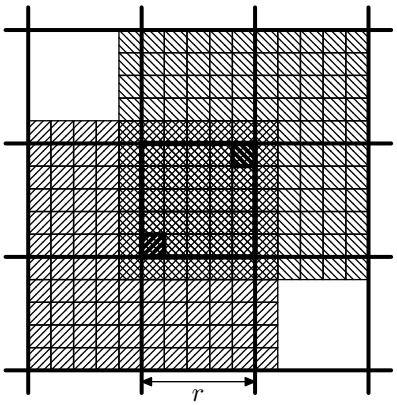
Liens avec le premier groupage

Le groupage carré correspond à la restriction aux automates cellulaires de dimension 1, de rayon 1 dans le cas où les seules transformations $\tilde{\mathsf{P}}\mathsf{CS}$ autorisées sont les transformations de la forme $\langle n, n, 0\rangle$. Nous montrons qu'en fait, dans le cas de la dimension 1, toute

classe d'équivalence du groupage généralisé (*i.e.* pour la relation $\sim$) contient une union non vide de classes d'équivalence du groupage carré (*i.e.* pour la relation $\sim_\square$). Autrement dit, le groupage carré définit un préordre plus fin que le groupage généralisé.

Proposition 1. *Pour tout automate cellulaire $\mathcal{A}$ de dimension* 1*, la classe d'équivalence de $\mathcal{A}$ contient une union non vide de classes d'équivalence du groupage carré (*i.e. *il existe un automate cellulaire $\mathcal{A}_\square$ de rayon de voisinage* 1 *tel que $\mathcal{A} \sim \mathcal{A}_\square$ et, pour tout automate cellulaire $\mathcal{B}$ de rayon de voisinage 1, si $\mathcal{B} \sim_\square \mathcal{A}_\square$ alors $\mathcal{B} \sim \mathcal{A}$).*

Démonstration. On montre d'abord que toute classe d'équivalence contient un automate cellulaire de rayon de voisinage 1. Soit $\mathcal{A}$ un automate cellulaire de rayon de voisinage r. Alors le transformé $\mathcal{A}^{\langle r_{\mathrm{id}},1,0\rangle}$ de $\mathcal{A}$ est un automate cellulaire de rayon de voisinage 1. la figure 2.14 démontre le principe de la construction pour un automate cellulaire de dimension 2, de rayon de voisinage 5. Notons au passage que cette propriété est vérifiée en toute dimension. De même, il existe un automate cellulaire unidirectionnel dans chaque classe d'équivalence, la construction est une variante de la précédente (prendre par exemple $\mathcal{A}^{\langle 2r_{\mathrm{id}},1,(-r,\ldots,-r)\rangle}$).



Les cellules hachurées correspondent, dans $\mathbb{Z}^2$, aux voisinages des deux cellules noires dans le cas d'un automate cellulaire de rayon de voisinage 5. Les traits épais délimitent les nouvelles cellules obtenues par découpage spatial. Les deux cellules noires étant extrêmes dans leur nouvelle cellule et ne dépendant que de cellules appartenant à des nouvelles cellules dans un voisinage de rayon 1, l'automate cellulaire obtenu après transformation est de rayon de voisinage 1.

FIG. 2.14 – Rayon de voisinage et transformations géométriques dans $\mathbb{Z}^2$

On montre ensuite que tous les transformés d'un automate cellulaire par une transformation $\tilde{\mathsf{PCS}}$ appartiennent à la même classe d'équivalence. Il s'agit ici d'une propriété de groupage,

$$\mathfrak{A} \vdash \forall x \forall \alpha \forall \beta \left(x^\alpha \preccurlyeq x^\beta \wedge x^\beta \preccurlyeq x^\alpha \right) \quad ,$$

conséquence du fait que $\preccurlyeq$ est une relation de préordre et de la propriété

$$\mathfrak{A} \vdash \forall x \forall \alpha \left(x \preccurlyeq x^\alpha \wedge x^\alpha \preccurlyeq x \right) \quad .$$

qui est immédiate. ■

Chaînes infinies incomparables

Nous montrons que l'ordre induit par la généralisation du groupage n'est pas trivial. En effet, il existe une infinité de classes d'équivalence et des chaînes infinies croissantes incomparables.

On définit la famille $\left(\boldsymbol{\xi}_{m,n}\right)_{m,n\in\mathbb{N}}$ d'automates cellulaires $\left(\mathbb{Z}^d, [\![1, m+n+1]\!], V^d_{\mathsf{Moore}}, \delta\right)$ dont la règle locale de transition est définie par

$$\delta\left(x_0 \ldots, x_{3^d-1}\right) = \begin{cases} x_0 & \text{si } \left|\{n \mid \exists i, x_i = n\}\right| = 1 \ ; \\ m+1 & \text{si } \left|\{n \mid \exists i, x_i = n\}\right| > 1 \text{ et } \min\{x_i\} \leqslant m \ ; \\ \max\{x_i\} & \text{sinon.} \end{cases}$$

Proposition 2. *Pour toute paire de couples d'entiers m, n et m', n', si $m' \leqslant m$ et $n' \leqslant n$ alors $\boldsymbol{\xi}_{m',n'} \leqslant \boldsymbol{\xi}_{m,n}$.*

Démonstration. Soient m, n et m', n' deux couples d'entiers tels que $m' \leqslant m$ et $n' \leqslant n$. L'automate cellulaire $\boldsymbol{\xi}_{m',n'}$ est un sous-automate de l'automate cellulaire $\boldsymbol{\xi}_{m,n}$. Pour s'en convaincre, on définit l'application injective φ de $[\![1, m'+n'+1]\!]$ dans $[\![1, m+n+1]\!]$ par $\varphi(i) = i - m' + m$. ■

Proposition 3. *Pour toute paire de couples d'entiers m, n et m', n', si $m' \neq m$ ou $n' \neq n$ alors $\boldsymbol{\xi}_{m',n'}$ et $\boldsymbol{\xi}_{m,n}$ appartiennent à des classes d'équivalence distinctes. De plus, si $m' < m$ et $n' > n$ ou si $m' > m$ et $n' < n$ alors $\boldsymbol{\xi}_{m',n'}$ et $\boldsymbol{\xi}_{m,n}$ sont incomparables.*

Démonstration. Considérons l'espace des phases de $\boldsymbol{\xi}_{m,n}$ restreint aux configurations périodiques. Les $m+n+1$ configurations monochromatiques sont des points fixes. Toute configuration périodique évolue en un nombre fini de pas vers un de ces points fixes. Si une configuration n'est pas monochromatique et si tous ses états appartiennent à l'ensemble $[\![1, m]\!]$, la configuration converge vers $\overline{m+1}$. Les autres configurations convergent vers la configuration monochromatique d'état l'état de valeur maximale de la configuration initiale. La figure 2.15 synthétise ces observations.

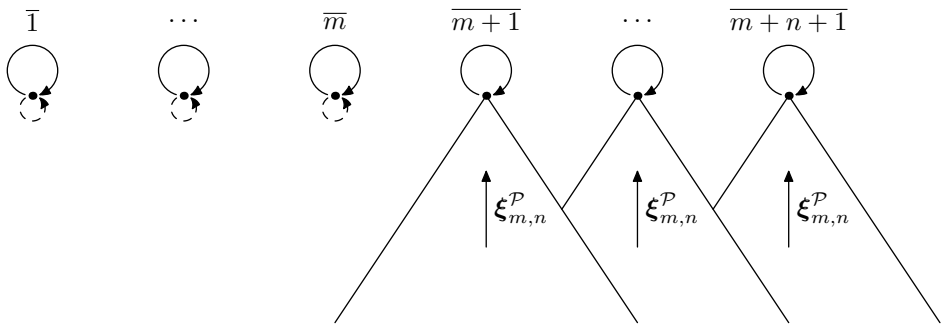


FIG. 2.15 – Espace des phases de $\boldsymbol{\xi}^{\mathcal{P}}_{m,n}$

Les transformations **PCS** conservant les points fixes et les composantes connexes de l'espace des phases, la transformation de $\boldsymbol{\xi}_{m,n}$ engendre un automate cellulaire dont l'espace des phases sur les configurations périodiques est toujours du type de la figure 2.15.

Soient maintenant m, n et m', n' deux couples d'entiers. Pour qu'un transformé de $\boldsymbol{\xi}_{m',n'}$ soit un sous-automate d'un transformé de $\boldsymbol{\xi}_{m,n}$, il faut donc au moins que $m' \leqslant m$ et $n' \leqslant n$. Ce qui conclut la démonstration. ■

La figure 2.16 synthétise la hiérarchie entre les automates cellulaires $\boldsymbol{\xi}_{m,n}$. Grâce à cette famille d'automates cellulaires on peut en particulier exhiber des chaînes infinies croissantes de classes d'équivalence telles que, si on choisit deux éléments sur des chaînes distinctes, ils sont incomparables (considérer par exemple les deux suites d'automates cellulaires $\left(\boldsymbol{\xi}_{i,0}\right)_{i\in\mathbb{N}}$ et $\left(\boldsymbol{\xi}_{0,j}\right)_{j\in\mathbb{N}}$).

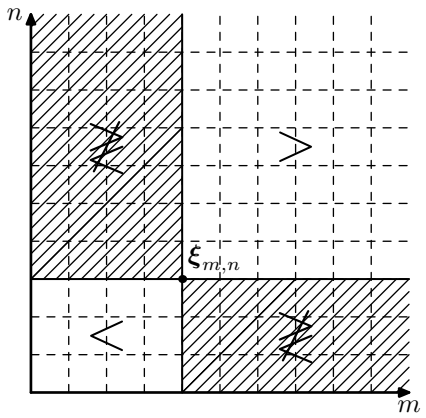


FIG. 2.16 – Comparaison des $\boldsymbol{\xi}_{m,n}$

Le bas de l'ordre

Comme dans le cas du groupage carré, on peut montrer que certaines classes d'équivalence du bas de l'ordre correspondent à des familles bien connues et intuitivement «simples» d'automates cellulaires.

Minimum global. L'automate cellulaire à un seul état, noté $\perp$, est seul dans sa classe d'équivalence et est le minimum global pour l'ordre induit.

Proposition 4. *La classe d'équivalence constituée de l'unique automate cellulaire à un état est un minimum global pour l'ordre.*

Démonstration. L'automate cellulaire à un seul état ne se transforme qu'en lui-même par une transformation **PCS**. De plus, il ne possède aucun autre sous-automate que lui-même. Il est donc seul dans sa classe d'équivalence. D'autre part, il est sous-automate de tout automate cellulaire possédant un état quiescent. Or, tout automate cellulaire contient parmi ses transformés temporels un automate cellulaire à état quiescent. ■

Automates nilpotents. L'ensemble des automates cellulaires nilpotents distincts de $\perp$ forme une classe d'équivalence située au bas de l'ordre : si un automate cellulaire différent de $\perp$ est simulé par un automate cellulaire nilpotent alors il est nilpotent.

Proposition 5. *La classe d'équivalence constituée des automates cellulaires nilpotents différents de $\perp$ est au bas de l'ordre.*

Démonstration. L'espace des phases d'un automate cellulaire nilpotent est consitué d'un unique point fixe atteint par toutes les autres configurations en un temps fini. Cette structure est une caractérisation des automates cellulaires nilpotents. Si T est la durée de la transitoire d'un automate cellulaire nilpotent, l'espace des phases du transformé

par C_T de cet automate cellulaire consiste en un point fixe atteint par toutes les autres configurations en un pas. D'autre part, l'application d'une transformation P à un tel automate cellulaire permet d'augmenter le nombre d'états en conservant cette structure. On en déduit que les automates cellulaires nilpotents se simulent entre eux et que tout automate cellulaire simulé par un automate cellulaire nilpotent est lui-même nilpotent. ■

Translations généralisées. L'ensemble des automates cellulaires périodiques à une translation près distincts de $\perp$ forme une classe d'équivalence située au bas de l'ordre.

Proposition 6. *La classe d'équivalence constituée des automates cellulaires périodiques à une translation près différents de $\perp$ est au bas de l'ordre.*

Démonstration. L'espace des phases d'un automate cellulaire périodique à une translation près est tel qu'il existe deux suites d'arêtes généralisées $\underbrace{\rightarrow \cdots \rightarrow}_{T}$ et

$$\dashrightarrow_{i_1} \cdots \dashrightarrow_{i_n} \dashleftarrow_{j_1} \cdots \dashleftarrow_{j_m}$$

telles que, pour tout sommet, les deux chemins partant de ce sommet étiquetés par ces suites d'arêtes atteignent un même sommet. Cette structure caractérise les automates cellulaires périodiques à une translation près. Si T est la période d'un tel automate cellulaire, l'espace des phases du transformé par C_T de cet automate cellulaire est une translation. D'autre part, l'application d'une transformation P et d'une transformation S à un tel automate cellulaire permet d'augmenter le nombre d'états et d'annuler la translation en conservant cette structure. On en déduit que les automates cellulaires périodiques à une translation près se simulent entre eux et que tout automate cellulaire simulé par un automate cellulaire périodique à une translation près est lui-même périodique à une translation près. ■

Le haut de l'ordre

Contrairement au cas du groupage carré, il existe une classe d'équivalence maximale dans le cadre du groupage généralisé. De même qu'on trouve au bas de l'ordre les automates cellulaires «simples», on trouve en haut de l'ordre les automates cellulaires qui produisent toutes les évolutions possibles : les automates cellulaires intrinsèquement universels.

Proposition 7. *La classe d'équivalence constituée des automates cellulaires intrinsèquement universels est un maximum global pour l'ordre.*

Démonstration. L'existence d'automates cellulaires intrinsèquement universels assure que l'ordre admet un maximum global. Pour montrer que la classe d'équivalence maximale est constituée uniquement des automates cellulaires intrinsèquement universels, on applique le théorème 12 au cas du groupage généralisé. ■

Au vu de ce résultat, on peut se demander si, de même qu'il existe des classes d'équivalence situées au bas de l'ordre, il en existe en haut de l'ordre, c'est-à-dire s'il existe des classes d'équivalence situées juste au-dessous des automates cellulaires intrinsèquement universels. Intuitivement, cette propriété n'est pas désirée car l'universalité est une sorte de comportement maximal limite. Nous allons montrer que la réalité est conforme à l'intuition.

Le groupage généralisé admet une notion de maximum local par le biais du produit cartésien d'automates. On en déduit qu'il ne peut exister au plus qu'une classe d'équivalence en haut de l'ordre.

Proposition 8. *Soient $\mathcal{A}$ et $\mathcal{B}$ deux automates cellulaires. Le produit cartésien $\mathcal{A} \times \mathcal{B}$ de ces automates est un maximum local : $\mathcal{A} \leqslant \mathcal{A} \times \mathcal{B}$ et $\mathcal{B} \leqslant \mathcal{A} \times \mathcal{B}$.*

Démonstration. Soient $\mathcal{A}$ et $\mathcal{B}$ deux automates cellulaires. Montrons que $\mathcal{A} \leqslant \mathcal{A} \times \mathcal{B}$, l'autre cas est symétrique. Il existe une durée T telle que le transformé par C_T de $\mathcal{B}$ admette un état quiescent q_0. Le transformé par C_T de $\mathcal{A} \times \mathcal{B}$ admet pour sous-automate le transformé par C_T de $\mathcal{A}$ par l'application injective $\varphi : Q_{\mathcal{A}} \to Q_{\mathcal{A}} \times Q_{\mathcal{B}}$ définie pour tout état q par $\varphi(q) = (q, q_0)$. ■

Corollaire 3. *Il existe au plus une classe d'équivalence en haut de l'ordre.*

Démonstration. Soient $\mathcal{A}$ et $\mathcal{B}$ deux automates cellulaire non intrinsèquement universels, montrons que leur produit cartésien $\mathcal{A} \times \mathcal{B}$ n'est pas intrinsèquement universel.

Supposons que $\mathcal{A} \times \mathcal{B}$ soit intrinsèquement universel et soit $\mathcal{U}$ un automate cellulaire intrinsèquement universel. Alors, par définition il existe une transformation **PCS** α telle que $\mathcal{U} \subseteq \mathcal{A}^\alpha \times \mathcal{B}^\alpha$. Soit φ l'application injective de $Q_{\mathcal{U}}$ dans $Q_{\mathcal{A}^\alpha} \times Q_{\mathcal{B}^\alpha}$ associée. On choisit pour $\mathcal{U}$ un automate cellulaire intrinsèquement universel d'un type un peu particulier (voir le chapitre 3 pour une explication du fonctionnement de ce type de construction) dont l'ensemble d'états se décompose en $Q_{\mathcal{U}} = Q \times \{a, b\} \cup \{a, b\}$. L'automate $\mathcal{U}$ se comporte comme une machine de Turing multi-têtes sur un espace de dimension d avec Q l'ensemble d'états des têtes et $\{a, b\}$ l'alphabet du ruban. On observe alors le codage de a et b dans $\mathcal{A}^\alpha \times \mathcal{B}^\alpha$, en particulier la projection de ce codage sur les ensembles d'états de $\mathcal{A}^\alpha$ et $\mathcal{B}^\alpha$. Il y a trois possibilités :

- Soit $\pi_1(\varphi(a)) = \pi_1(\varphi(b))$ et $\pi_2(\varphi(a)) \neq \pi_2(\varphi(b))$. Dans ce cas, la première composante du codage, dirigée par $\mathcal{A}^\alpha$ ne distingue pas a de b et en conséquence ne peut réaliser aucun calcul. On en conclut que c'est la deuxième composante qui calcule et donc que l'application $\psi : Q_{\mathcal{U}} \to Q_{\mathcal{B}^\alpha}$ qui, à tout état q, associe $\pi_2(\varphi(q))$ est injective et que $\mathcal{U} \leqslant \mathcal{B}$.
- Soit $\pi_1(\varphi(a)) \neq \pi_1(\varphi(b))$ et $\pi_2(\varphi(a)) = \pi_2(\varphi(b))$. Ce cas est symétrique du précédent. On en déduit que $\mathcal{U} \leqslant \mathcal{A}$.
- Soit $\pi_1(\varphi(a)) \neq \pi_1(\varphi(b))$ et $\pi_2(\varphi(a)) \neq \pi_2(\varphi(b))$. Dans ce cas, les deux composantes du codage distinguent a et b et en conséquence, pour préserver l'injectivité du codage au cours du calcul, chaque composante doit être capable de réaliser tous les calculs. Les deux composantes étant indépendantes, on en déduit que $\mathcal{U} \leqslant \mathcal{A}$ et $\mathcal{U} \leqslant \mathcal{B}$.

D'après les hypothèses, ni $\mathcal{A}$ ni $\mathcal{B}$ ne sont intrinsèquement universels. On arrive donc à une contradiction dans chacun des trois cas. ■

Le résultat précédent peut s'interpréter de la manière suivante :

1. soit pour tout automate cellulaire non intrinsèquement universel $\mathcal{A}$ il existe un automate cellulaire non intrinsèquement universel $\mathcal{B}$ tel que $\mathcal{A} < \mathcal{B}$,
2. soit il existe un automate cellulaire non intrinsèquement universel $\mathcal{A}$ tel que pour tout automate cellulaire non intrinsèquement universel $\mathcal{B}$, on a $\mathcal{B} \leqslant \mathcal{A}$.

On veut montrer que le second cas, l'existence d'un automate cellulaire en haut de l'ordre, n'est pas réalisable. Pour cela, on relie sa réalisation à un problème de décision qui sera montré indécidable dans le chapitre suivant, concluant ainsi la démonstration.

Proposition 9. *S'il existe une classe d'équivalence au haut de l'ordre, alors le problème de décision* CA-UNIV *est décidable.*

CA-UNIV	
Entrée	Un automate cellulaire $\mathcal{A}$
Question	L'automate $\mathcal{A}$ est-il intrinsèquement universel ?

Démonstration. Supposons qu'il existe une classe d'équivalence en haut de l'ordre et soit $\mathcal{U}_{-1}$ un automate cellulaire appartenant à cette classe. Soit $\mathcal{U}$ un automate cellulaire intrinsèquement universel. Montrons que l'intrinsèque universalité est décidable. Soit $\mathcal{A}$ un automate cellulaire.

Si $\mathcal{A}$ est intrinsèquement universel, il existe une transformation **PCS** α telle que $\mathcal{U} \subseteq \mathcal{A}^\alpha$. Par énumération des transformations **PCS**, la transformation d'automates cellulaires et le test sous-automate étant calculables, si $\mathcal{A}$ est intrinsèquement universel, alors on l'observe en un temps fini.

Si $\mathcal{A}$ n'est pas intrinsèquement universel, alors $\mathcal{A} \leqslant \mathcal{U}_{-1}$. Il existe deux transformations **PCS** α et β telles que $\mathcal{A}^\alpha \subseteq \mathcal{U}_{-1}^\beta$. Pour les mêmes raisons que ci-dessus, si $\mathcal{A}$ n'est pas intrinsèquement universel, alors on l'observe en un temps fini. ■

Problèmes de décision associés

Nous montrons que la relation de simulation du groupage généralisé est indécidable. En pratique, nous montrons l'indécidabilité de deux problèmes plus difficiles.

Proposition 10. *Le problème de décision* CA-LEFT-POW *est indécidable.*

CA-LEFT-POW	
Entrée	Deux automates cellulaires $\mathcal{A}$ et $\mathcal{B}$
Question	Existe-t-il une transformation $\tilde{\mathbf{P}}\mathbf{CS}$ α telle que $\mathcal{A}^\alpha \subseteq \mathcal{B}$?

Démonstration. Un automate cellulaire $\mathcal{A}$ est nilpotent si et seulement si il existe une durée T telle que l'application $G_{\mathcal{A}}^T$ est constante. Cette dernière propriété est équivalente à l'existence d'une transformation **C** α telle que $\mathcal{A}^\alpha$ est constante. Soit $\mathcal{B}$ un automate cellulaire constant possédant autant d'états que $\mathcal{A}$. Alors $\mathcal{A}$ est nilpotent si et seulement si il existe une transformation $\tilde{\mathbf{P}}\mathbf{CS}$ α telle que $\mathcal{A}^\alpha \subseteq \mathcal{B}$. La nilpotence des automates cellulaires étant indécidable, le problème CA-LEFT-POW l'est aussi. ■

Proposition 11. *Le problème de décision* CA-RIGHT-POW *est indécidable.*

CA-RIGHT-POW	
Entrée	Deux automates cellulaires $\mathcal{A}$ et $\mathcal{B}$
Question	Existe-t-il une transformation $\tilde{\mathbf{P}}\mathbf{CS}$ α telle que $\mathcal{A} \subseteq \mathcal{B}^\alpha$?

Démonstration. Soit $\mathcal{A}$ un automate cellulaire intrinsèquement universel. Un automate cellulaire $\mathcal{B}$ est intrinsèquement universel si et seulement si il existe une transformation $\tilde{\mathbf{P}}\mathbf{CS}$ α telle que $\mathcal{A} \subseteq \mathcal{B}^\alpha$. Le problème CA-UNIV étant indécidable (démonstration dans le chapitre suivant), le problème CA-RIGHT-POW l'est aussi. ■

2.4.4 Familles classiques d'automates cellulaires

Un des intérêts du groupage est que de nombreuses familles classiques d'automates cellulaires ont un comportement identifiable dans ce cadre.

Classes d'équivalences. Certaines familles d'automates cellulaires correspondent à des classes d'équivalence du groupage généralisé. Ainsi, nous avons vu dans les paragraphes précédents :
- l'automate cellulaire à un seul état,
- les automates cellulaires nilpotents,
- les automates cellulaires périodiques à une translation près,
- les automates cellulaires intrinsèquement universels.

Stabilité par simulation. D'autres familles d'automates cellulaires vérifient la propriété suivante : tout automate cellulaire simulé par un automate cellulaire de la famille appartient à cette famille. C'est le cas, par exemple, des automates cellulaires injectifs et des automates cellulaires surjectifs.

Proposition 12. *Tout automate cellulaire simulé par un automate cellulaire injectif est injectif.*

Démonstration. Soit $\mathcal{A}$ un automate cellulaire. Pour toute transformation **PCS** α, l'automate cellulaire transformé $\mathcal{A}^\alpha$ est injectif si et seulement si $\mathcal{A}$ l'est car les ensembles de configurations de ces deux automates cellulaires sont en bijection et que la dite bijection est compatible avec l'application de la règle globale de transition. De plus, tout sous-automate d'un automate cellulaire injectif est un automate cellulaire injectif. On en déduit que tout automate cellulaire simulé par un automate cellulaire injectif est lui-même injectif. ■

Proposition 13. *Tout automate cellulaire simulé par un automate cellulaire surjectif est surjectif.*

Démonstration. Le raisonnement est le même que pour la proposition précédente mais on utilise la variante faible du théorème de Moore-Myhill qui caractérise les automates cellulaires surjectifs par l'injectivité sur les configurations finies d'une de leurs puissances. On en déduit que tout automate cellulaire simulé par un autoamte cellulaire surjectif est lui-même surjectif. ■

Majoration. Pour d'autres familles d'automates cellulaires on peut montrer que tout automate cellulaire est simulé par un automate cellulaire de la famille. En particulier, cela implique que cette famille contient des automates cellulaire intrinsèquement universels.

Proposition 14. *Tout automate cellulaire est simulé par un automate cellulaire unidirectionnel.*

Démonstration. La preuve se déduit de la construction donnée dans la démonstration de la proposition 1. ■

Un automate cellulaire totalistique est un automate cellulaire dont l'ensemble d'états est inclus dans $\mathbb{N}$ et dont la règle locale de transition est du type

$$\delta(x_1, \ldots, x_n) = f\left(\sum_{i=1}^{n} x_i\right) \quad .$$

Proposition 15. *Tout automate cellulaire est simulé par un automate cellulaire totalistique.*

Démonstration. Pour montrer cette proposition, il suffit de montrer que tout automate cellulaire unidirectionnel est simulé par un automate cellulaire totalistique. Soit $\mathcal{A}$ un automate cellulaire unidirectionnel de dimension d et d'ensemble d'états $[\![0, n-1]\!]$. On définit l'automate cellulaire totalistique $\mathcal{B}$ de la manière suivante. Son ensemble d'états est $[\![0, n^{(3r)^d} - 1]\!]$. L'idée est de coder une configuration C de $\mathcal{A}$ en une configuration C' de $\mathcal{B}$ définie en toute position $p = (p_0, \ldots, p_{d-1})$ par :

$$C'_p = n^{(p_0 \bmod 3)+3(p_1 \bmod 3)+\ldots+3^{d-1}(p_{d-1} \bmod 3)} C_p \quad .$$

Ainsi, en toute position p, la somme des voisins écrite en base n est un nombre dont 2^d chiffres sont non-nuls. La position de ces chiffres dans l'écriture permet de retrouver la position de chaque voisin. ■

Proposition 16. *Tout automate cellulaire est simulé par un automate cellulaire polynomial.*

Démonstration. Soit $\mathcal{A}$ un automate cellulaire à n états. Si n est premier, alors, si on regarde l'ensemble d'états de $\mathcal{A}$ comme le corps fini $F(p)$, la règle globale de transition δ de $\mathcal{A}$ est polynomiale, de polynôme

$$P(X_1, \ldots, X_k) = \sum_{(x_1,\ldots,x_k) \in Q_{\mathcal{A}}^k} \prod_{i=1}^{k} \prod_{x \neq x_i} \frac{X_i - x}{x_i - x} \delta_{\mathcal{A}}(x_1, \ldots, x_k) \quad .$$

■

Un automate cellulaire conservatif est un automate cellulaire $\mathcal{A}$ d'ensemble d'états inclus dans $\mathbb{Z}$ et tel que, pour toute configuration périodique C de période π :

$$\sum_{p \in [\![0,\pi-1]\!]^d} C_p = \sum_{p \in [\![0,\pi-1]\!]^d} G_{\mathcal{A}}(C)_p \quad .$$

Proposition 17. *Tout automate cellulaire est simulé par un automate cellulaire conservatif.*

Démonstration. Soit $\mathcal{A}$ un automate cellulaire à n états. On définit l'automate cellulaire conservatif $\mathcal{B}$ qui simule $\mathcal{A}$ de la manière suivante. L'ensemble d'états de $\mathcal{B}$ est $[\![-n, n]\!]$. L'automate cellulaire $\mathcal{B}$ simule $\mathcal{A}$ par groupage spatial de deux cellules. Un état q de $\mathcal{A}$ est codé par une paire $(q, -q)$ d'états de $\mathcal{B}$. Le voisinage de $\mathcal{B}$ est le voisinage de $\mathcal{A}$ transformé par la transformation précédente, auquel on ajoute ce qu'il faut pour qu'une cellule puisse observer ses voisines normales pour $\mathcal{A}$ ainsi que ce que voit sa voisine dans le groupage spatial. Si le codage est localement correct, on effectue une transition, sinon on conserve son propre état inchangé. ■

2.4.5 Structure

Le groupage généralisé induit une relation de préordre. Cette structure permet de montrer des résultats plus facilement, en se reposant sur certaines propriétés structurelles comme la transitivité, par exemple. Nous montrons cependant que la relation de simulation induite par le groupage généralisé n'admet pas de structure de demi-treillis.

Translations pures. On se place en dimension 1 et on considère la famille $\left(\boldsymbol{\sigma}_v^{(z)}\right)_{v\in\mathbb{Z}}$ des automates cellulaires de translation pure à z états :

$$\boldsymbol{\sigma}_v^{(z)} = (\mathbb{Z}, [\![1, z]\!], \{-v\}, q \mapsto q) \quad .$$

Transformés. L'image par une transformation $\tilde{\mathsf{P}}\mathsf{CS'}$ $\langle m, mn, mk\rangle$ d'un automate cellulaire $\boldsymbol{\sigma}_v^{(z)}$ est l'automate cellulaire

$$o_{-m} \circ \sigma_{mk} \circ \left(\boldsymbol{\sigma}_v^{(z)}\right)^{mn} \circ o_m = \boldsymbol{\sigma}_{nv+k}^{(z^m)}$$

Sous-automates. Les sous-automates d'un automate cellulaire $\boldsymbol{\sigma}_v^{(z)}$ sont exactement les automates cellulaires $\boldsymbol{\sigma}_v^{(z')}$ pour $z' < z$. En particulier, pour $z' = 1$ cet automate cellulaire atteint le minimum du groupage.

Produits cartésiens. Les produits cartésiens d'automates cellulaires de translations pures sont tous dans la classe d'équivalence d'un automate cellulaire de type $\prod_{i=1}^{l} \boldsymbol{\sigma}_{v_i}$ où la suite des (v_i) est strictement croissante et $\boldsymbol{\sigma}_v$ désigne l'automate cellulaire de translation pure à 2 états $\boldsymbol{\sigma}_v^{(2)}$. L'image par une transformation $\tilde{\mathsf{P}}\mathsf{CS'}$ $\langle m, mn, mk\rangle$ de l'automate cellulaire produit $\prod_{i=1}^{l} \boldsymbol{\sigma}_{v_i}$ est $\prod_{i=1}^{l} \boldsymbol{\sigma}_{nv_i+k}^{(2^m)}$. Les sous-automates de ce dernier automate cellulaire produit sont exactement les automates cellulaires $\prod_{i=1}^{l'} \boldsymbol{\sigma}_{v'_i}^{(k'_i)}$ où la suite $(v'_1, \ldots, v'_{l'})$ est une sous-suite de $(nv_1 + k, \ldots, nv_l + k)$ et $k'_i \in [\![2, 2^m]\!]$.

Représentant canonique. Les produits cartésiens d'un seul ou de deux automates cellulaires de translations pures sont seuls, chacun dans sa classe d'équivalence. À tout produit d'au moins trois automates cellulaires de translation pure $\prod_{i=1}^{l} \boldsymbol{\sigma}_{v_i}$, on associe la suite croissante de $l-2$ éléments de $\mathbb{Q} \cap]1, +\infty[$

$$\left(\frac{v_3 - v_1}{v_2 - v_1}, \ldots, \frac{v_l - v_1}{v_2 - v_1}\right) \quad .$$

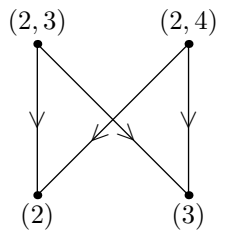
Cette suite est laissée invariante par transformation $\tilde{\mathsf{P}}\mathsf{CS'}$. Soit $(\alpha_1, \ldots, \alpha_l)$ une telle suite. On considère un sous-automate d'un automate cellulaire représenté par cette suite. Si le sous-automate a autant de composantes, il admet la même suite. Si le sous-automate a une composante de moins, la suite associée au sous-automate peut être de trois types :

- $(\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_l)$: retrait d'un vecteur v_j avec $j > 2$,
- $\left(\frac{\alpha_2}{\alpha_1}, \ldots, \frac{\alpha_l}{\alpha_1}\right)$: retrait du vecteur v_2,
- $\left(\frac{\alpha_2 - 1}{\alpha_1 - 1}, \ldots, \frac{\alpha_l - 1}{\alpha_1 - 1}\right)$: retrait du vecteur v_1.

Ces suites croissantes de $(\mathbb{Q} \cap]1, +\infty[)^*$ sont donc en bijection avec les classes d'équivalence des produits cartésiens d'au moins 3 automates cellulaires de translation pure. De plus, nous connaissons complètement leurs sous-automates.

Proposition 18. *La relation de simulation induite par le groupage généralisé n'induit pas de structure de demi-treillis.*

Démonstration. On considère les classes d'équivalence associées aux suites (2), (3), $(2, 3)$ et $(2, 4)$. De la discussion précédente, on déduit les relations suivantes :



De plus, pour chacune de ces inégalités, il n'existe aucune classe d'équivalence strictement comprise entre les deux classes. La relation d'ordre induit $\leqslant$ n'admet donc ni borne supérieure, ni borne inférieure. ■

2.5 Un groupage fortement structuré

La première esquisse de groupage présentée dans les paragraphes précédents est une généralisation intéressante du groupage carré. En particulier, l'ordre induit par ce groupage admet une classe d'équivalence maximale qui correspond aux automates cellulaires intrinsèquement universels.

Cependant, le produit cartésien de certains automates cellulaires engendre une structure compliquée qui ne permet pas d'avoir toutes les propriétés structurelles intéressantes d'un demi-treillis. Nous modifions le groupage généralisé de sorte que l'ordre induit admette bien une structure de demi-treillis où l'opérateur naturel de borne supérieure correspond au produit cartésien.

2.5.1 Transformations géométriques généralisées

Transformation. Intuitivement, une transformation de nature géométrique sur les diagramme espace-temps est maintenant une transformation géométrique au sens précédent mais dans laquelle on considère une famille finie de diagrammes espace-temps de départ. Nous proposons la formalisation suivante.

Définition 5. *Une* transformation géométrique *est un triplet* $(k, l, \mathbf{\Lambda})$ *où* k *et* l *sont des entiers strictement positifs et* $\mathbf{\Lambda}$ *une application*

$$\mathbf{\Lambda} : \mathbb{N} \times \mathbb{Z}^d \longrightarrow \left([\![1, l]\!] \times \mathbb{N} \times \mathbb{Z}^d\right)^k$$

Mise-en-oeuvre. La transformation de diagrammes espace-temps sur l'alphabet Q par une transformation géométrique $(k, l, \mathbf{\Lambda})$ est définie par l'application $\overline{\mathbf{\Lambda}}_Q$ de $\left(Q^{\mathbb{N}\times\mathbb{Z}^d}\right)^l$ dans $\left(Q^k\right)^{\mathbb{N}\times\mathbb{Z}^d}$ définie pour tout l-uple de diagrammes espace-temps $\Delta = (\Delta_i)_{i\in[\![1,l]\!]}$ sur Q, en tout temps $t \in \mathbb{N}$ et en toute position $p \in \mathbb{Z}^d$ par

$$\overline{\mathbf{\Lambda}}_Q(\Delta)(t, p) = (\Delta(\mathbf{\Lambda}(t, p)_1), \ldots, \Delta(\mathbf{\Lambda}(t, p)_k)) \quad .$$

Pseudo-composition. On définit la composition de deux transformations géométriques en observant l'application successive de ces transformations sur un diagramme espace-temps. Soit $(k, l, \mathbf{\Lambda})$ et $(k', l', \mathbf{\Lambda}')$ deux transformations géométriques. L'application successive de ces deux transformations engendre des diagrammes espace-temps sur $\left(Q^k\right)^{k'}$.

Puisqu'on travaille toujours à un renommage des états près, on peut considérer ces diagrammes espace-temps comme étant à valeurs dans $Q^{kk'}$ et étant engendrés par application de la transformation $(kk', ll', \mathbf{\Lambda}' \circ \mathbf{\Lambda})$ définie en tout temps $t \in \mathbb{N}$ et en toute position $p \in \mathbb{Z}^d$ par

$$(\mathbf{\Lambda}' \circ \mathbf{\Lambda})(t,p) = \Big(\mathbf{\Lambda}\left(\mathbf{\Lambda}'(t,p)_1\right)_1, \ldots, \mathbf{\Lambda}\left(\mathbf{\Lambda}'(t,p)_{k'}\right)_k\Big) \quad .$$

► Les notations pour le choix des diagrammes espace-temps de départ sont un peu simplifiées pour ne pas compliquer les expressions.

L'opération correspondant – à un renommage près – à la composition sur l'application des tranformations, la notation composition est pertinente et a toutes les propriétés de la composition, en particulier la transitivité.

Pré-images. On introduit l'application $\widetilde{\mathbf{\Lambda}}$, qui à un ensemble de cellules d'un diagramme espace-temps associe l'ensemble des cellules qu'elles représentent.

$$\begin{array}{rccl} \widetilde{\mathbf{\Lambda}}: & 2^{\mathbb{N}\times\mathbb{Z}^d} & \longrightarrow & 2^{[\![1,l]\!]\times\mathbb{N}\times\mathbb{Z}^d} \\ & X & \longmapsto & \displaystyle\bigcup_{(t,p)\in X} \{\mathbf{\Lambda}(t,p)_1, \ldots, \mathbf{\Lambda}(t,p)_k\} \end{array}$$

Transformations sympathiques. Les transformations sympathiques sont définies de manière identique au cas précédent, à quelques détails près.

Définition 6. *Une transformation géométrique* $(k, l, \mathbf{\Lambda})$ *est une* transformation géométrique sympathique *si elle vérifie :*

1. *pour tout automate cellulaire* $\mathcal{A}$*, il existe un automate cellulaire* $\mathcal{B}$ *vérifiant*

$$\left\{\overline{\mathbf{\Lambda}}_{Q_{\mathcal{A}}}(\Delta)\right\}_{\Delta\in\mathrm{Diag}(\mathcal{A})^l} = \mathrm{Diag}(\mathcal{B})$$

2. *pour tout temps* $t \in \mathbb{N}$, $\widetilde{\mathbf{\Lambda}}(\{t+1\}\times\mathbb{Z}^d) \not\subseteq \widetilde{\mathbf{\Lambda}}(\{t\}\times\mathbb{Z}^d)$.

Transformations PCST. Les transformations **PCST** correspondent à la généralisation des transformations **PCS**. Une transformation $\mathbf{PCST}_{(F_i,v_i,T_i,s_i)_{i\in[\![1,l]\!]}}$ transforme un automate cellulaire $\mathcal{A}$ en l'automate cellulaire

$$\left(o^{-1}_{F_1,v_1} \circ \sigma_{s_1} \circ G^{T_1}_{\mathcal{A}} \circ o_{F_1,v_1}\right) \times \cdots \times \left(o^{-1}_{F_l,v_l} \circ \sigma_{s_l} \circ G^{T_l}_{\mathcal{A}} \circ o_{F_l,v_l}\right) \quad .$$

Caractérisation des transformations sympathiques

Ici encore, à cause de la nature très contrainte des automates cellulaires, les transformations **PCST** correspondent exactement aux transformations sympathiques.

Théorème 15. *Une transformation géométrique est sympathique si et seulement si elle s'exprime comme une transformation* **PCST**.

Démonstration. La démonstration procède ainsi. On commence par remarquer que toute cellule du diagramme espace-temps d'arrivée d'une transformation sympathique doit recevoir autant de cellules de chaque automate cellulaire de départ. Puis, on raisonne comme pour la démonstration originelle, de manière indépendante sur chaque diagramme espace-temps de départ. On conclut que la transformation est un produit cartésien de transformations **PCS**, c'est-à-dire une transformation **PCST**. ■

2.5.2 Axiomatisation

Les problèmes rencontrés avec les transformations **PCS** sont bien entendu toujours présents dans le cas des transformations **PCST**. Aussi, on introduit les transformations **P̃CST** et **P̃CST'** sur le même modèle que les transformations **P̃CS** et **P̃CS'**.

On propose de définir un modèle du groupage abstrait reposant sur les transformations **P̃CST** et la relation sous-automate de la manière suivante :

Obj est l'ensemble des automates cellulaires de dimension d ;

Trans est l'ensemble des transformations **P̃CST'** de dimension d ;

apply est l'opérateur de transformation des automates cellulaires ;

divide est la relation sous-automate ;

combine est la composition des transformations **P̃CST'** (*i.e.* **combine**$(\mathcal{T}_1, \mathcal{T}_2) = \mathcal{T}_2 \circ \mathcal{T}_1$).

Tous les axiomes sont vérifiés en utilisant exactement les mêmes arguments que pour l'axiomatisation précédente à base de transformations **P̃CS**.

2.5.3 Exploration

Liens avec le groupage précédent

Le groupage généralisé est un préordre plus fin que le nouveau groupage généralisé.

Proposition 19. *La relation de simulation précédente est plus fine que la nouvelle relation de simulation.*

Démonstration. Les transformations **PCS** étant des transformations **PCST** valides, les classes d'équivalence du groupage défini sur les transformations **PCST** sont des unions de classes d'équivalence du groupage défini par les transformations **PCS**. ■

Demi-treillis et idéaux

Le préodre induit par le nouveau groupage généralisé induit une structure de sup-demi-treillis sur l'ensemble des automates cellulaires.

Proposition 20. *Le produit cartésien définit une borne supérieure pour le groupage généralisé.*

Démonstration. Soient $\mathcal{A}$ et $\mathcal{B}$ deux automates cellulaires. Comme démontré dans le cas des transformations **PCS**, $\mathcal{A} \leqslant \mathcal{A} \times \mathcal{B}$ et $\mathcal{B} \leqslant \mathcal{A} \times \mathcal{B}$. Montrons que pour tout automate cellulaire $\mathcal{E}$ majorant $\mathcal{A}$ et $\mathcal{B}$, nécessairement $\mathcal{A} \times \mathcal{B} \leqslant \mathcal{E}$. Soit $\mathcal{E}$ un majorant de $\mathcal{A}$ et $\mathcal{B}$. Il existe quatre transformations **PCST** α, β, γ, δ telles que $\mathcal{A}^\alpha \subseteq \mathcal{E}^\beta$ et $\mathcal{B}^\gamma \subseteq \mathcal{E}^\delta$. On en déduit les deux relations : $\mathcal{A}^{\alpha\cdot\varphi(\mathcal{B},\gamma)} \subseteq \mathcal{E}^{\beta\cdot\varphi(\mathcal{B},\gamma)}$ et $\mathcal{B}^{\alpha\cdot\varphi(\mathcal{B},\gamma)} \subseteq \mathcal{E}^{\delta\cdot\psi(\mathcal{B},\alpha,\gamma)}$. La relation sous-automate et le produit cartésien étant compatibles d'une part et le produit cartésien et la transformation étant compatibles d'autre part, on en déduit : $(\mathcal{A} \times \mathcal{B})^{\alpha\cdot\varphi(\mathcal{B},\gamma)} \subseteq \mathcal{E}^{\beta\cdot\varphi(\mathcal{B},\gamma)} \times \mathcal{E}^{\delta\cdot\psi(\mathcal{B},\alpha,\gamma)}$, ce qui implique $\mathcal{A} \times \mathcal{B} \leqslant \mathcal{E}$ car il existe une transformation ζ telle que $\mathcal{E}^\zeta \cong \mathcal{E}^{\beta\cdot\varphi(\mathcal{B},\gamma)} \times \mathcal{E}^{\delta\cdot\psi(\mathcal{B},\alpha,\gamma)}$. ■

Cette nouvelle structure sur l'ensemble des automates cellulaires permet de caractériser certaines familles d'automates cellulaires comme des idéaux, *i.e.* des ensembles de classes d'équivalence stables par l'opération de borne supérieur et par passage aux classes inférieurs. Un idéal du groupage correspond donc à un ensemble d'automates cellulaires

stable par : transformation $\tilde{\mathsf{P}}\mathsf{CST}$, passage au sous-automate et produit cartésien de deux éléments de l'idéal. On distingue les idéaux principaux, qui contiennent une borne supérieure, des idéaux non-principaux.

Nous avons déjà rencontré plusieurs familles d'automates cellulaires dont les classes d'équivalence forment des idéaux dans les paragraphes précédents. Ainsi, le caractère principal de l'idéal des classes d'équivalence des automates cellulaires non intrinsèquement universels a été relié à la décidabilité de l'universalité intrinsèque. Nous présentons quelques autres idéaux correspondant à des familles classiques d'automates cellulaires.

Proposition 21. *L'idéal des classes d'équivalence des automates cellulaires réversibles est principal.*

Démonstration. Les automates cellulaires réversibles sont les automates cellulaires injectifs. Montrons tout d'abord que cette famille d'automates cellulaires correspond à un idéal. L'injectivité est préservée par les transformation $\tilde{\mathsf{P}}\mathsf{CST}$, par le passage au sous-automate et par produit cartésien. La principalité de cet idéal provient de l'existence d'automates cellulaires réversibles intrinsèquement universels pour les automates cellulaires réversibles, cette idée est discutée dans le chapitre consacré à l'universalité. ■

Proposition 22. *Dans le cas des dimensions supérieures à* 2*, l'idéal des classes d'équivalence des automates cellulaires surjectifs n'est pas principal.*

Démonstration. Les automates cellulaires surjectifs étant les automates cellulaires qui possèdent un transformé injectif sur les configurations finies, leurs classes d'équivalence forment un idéal, comme dans le cas des automates cellulaires injectifs. Montrons que si cet idéal est principal, alors la surjectivité des automates cellulaires est décidable. Supposons que l'idéal soit principal. Soit $\mathcal{A}$ un automate cellulaire. Alors, si $\mathcal{U}$ est un élément principal et si $\mathcal{A}$ est surjectif, $\mathcal{U}$ simule $\mathcal{A}$, ce qui peut se détecter en un temps fini. Si $\mathcal{A}$ n'est pas surjectif, alors, d'après les travaux de Moore et Myhill, il existe un motif fini d'Eden, c'est-à-dire n'apparaissant pas dans l'évolution de $\mathcal{A}$, ce qui peut se détecter en un temps fini. On conclut que, dans le cas des dimensions supérieures à 2, l'idéal correspondant aux automates cellulaires surjectifs n'est pas principal car la surjectivité des automates cellulaires est indécidable. ■

CHAPITRE

3 UNIVERSALITÉS

Dans ce dernier chapitre, on s'intéresse aux différentes formes d'universalité et en particulier à l'universalité intrinsèque. Après avoir présenté rapidement la problématique, on démontre que la propriété d'être intrinsèquement universel est indécidable et on exhibe de petits automates cellulaires intrinsèquement universels : un automate cellulaire intrinsèquement universel bilinéaire de dimension 1 *à* 2 *états et grand voisinage ainsi qu'un automate cellulaire intrinsèquement universel de dimension* 1 *à* 6 *états et voisinage de von Neumann.*

Sommaire

3.1 Différentes notions d'universalité 87
3.1.1 Universalités 87
3.1.2 Un résultat étonnant 88
3.1.3 Prototypes ayant de bonnes propriétés 89
3.1.4 Minimisation 90
3.2 Indécidabilité de l'universalité intrinsèque 91
3.2.1 Un automate cellulaire intrinsèquement universel 91
3.2.2 Automates cellulaires chaudières 92
3.2.3 Résultat 93
3.3 Un automate universel bilinéaire à deux états 94
3.4 Un automate universel à 6 états 97
3.4.1 Un automate cellulaire universel simple à 8 états 97
Considérations macroscopiques 98
Codage microscopique 99
3.4.2 Affinement du nombre d'états requis 102
Émuler la copie : de 8 à 7 états 102
Séparer les opérateurs : de 7 à 6 états 102

3.1 Différentes notions d'universalité

3.1.1 Universalités

Dès la construction historique par J. von Neumann [80] est présente une certaine forme d'automate cellulaire universel. En effet, J. von Neumann, pour construire son automate cellulaire auto-reproducteur, introduit dans les configurations et l'évolution de son automate cellulaire une machine de Turing universelle pour le calcul. En un certain sens, qui reste toujours à formaliser, cet automate cellulaire est universel pour le calcul ou Turing universel : il est possible, pour tout calcul Turing, de coder la configuration initiale de la machine de Turing en une configuration de l'automate cellulaire et, après avoir fait évoluer cette configuration par le biais de la règle globale de transition de l'automate cellulaire en question, de décoder la configuration finale de la machine de Turing, représentant le résultat du calcul, à partir de la configuration de l'automate cellulaire. Cette notion d'automate cellulaire Turing universel est très difficile à formaliser, parce qu'il faut

définir les fonctions de codage, de décodage et de détection d'arrêt valides, parce q'il faut transformer des objets finis en objets infinis, mais aussi parce que la notion de machine de Turing universelle ne possède pas non plus de définition formelle. S'il est facile de se convaincre de la validité d'une construction et ainsi de l'universalité Turing d'un automate cellulaire donné, il est plus difficile de décider de la non universalité Turing d'un automate cellulaire.

Cependant, les travaux de J. von Neumann [80] laissent aussi apparaître les prémisses d'une autre notion d'universalité pour les automates cellulaires : l'universalité intrinsèque. Un automate cellulaire intrinsèquement universel est un automate cellulaire qui peut reproduire, à un codage raisonnable de ses diagrammes espace-temps près, les diagrammes espace-temps de n'importe quel automate cellulaire. Cette notion a ensuite été utilisée avec succès de plus en plus formellement dans les travaux de E. R. Banks [20], puis J. Albert et K. Čulik II [34], la première définition apparaissant explicitement dans [54]. Le grand intérêt de cette notion est d'être formalisable simplement, les codages utilisés dans la littérature correspondent aux transformations $\tilde{\mathsf{P}}\mathsf{CS}$, et de permettre la démonstration de la non universalité de certains automates cellulaires. Une discussion détaillée sur la formalisation des deux notions d'universalité, leurs liens ainsi que les avantages de l'universalité intrinsèque est présentée par B. Durand et Zs. Róka [66].

Indépendamment de ces deux notions d'universalité, l'universalité des automates cellulaires a été étudiée dans le cadre restreint des automates cellulaires réversibles. Ainsi, certains automates cellulaires réversibles Turing universels ont été construits, voir par exemple les travaux de K. Morita [57]. En ce qui concerne la simulation au sens du groupage, une réinterprétation de certains résultats, en particulier les travaux de J. Durand-Lose [60], [9] ou encore K. Imai et K. Morita [75], met en avant l'existence d'automates cellulaires réversibles intrinsèquement universels pour la classes des automates cellulaires réversibles, c'est-à-dire capables de simuler, au même sens que pour les automates cellulaires intrinsèquement universels, tout automate cellulaire réversible.

3.1.2 Un résultat étonnant

En combinant les résultats de I. Rapaport [12], en particulier l'existence de chaînes infinies non-bornées qui implique l'absence de maximum global pour le groupage carré, avec l'existence d'un maximum global pour le groupage généralisé correspondant aux automates cellulaires intrinsèquement universels, nous obtenons le résultat surprenant suivant :

Corollaire 4. *Il n'existe pas d'automate cellulaire intrinsèquement universel simulant tous les automates cellulaires en temps réel, c'est-à-dire en se restreignant aux transformations du groupage carré.*

De plus, il est très facile, pour tout $\epsilon > 0$ de construire un automate cellulaire de dimension 1 qui simule tout automate cellulaire par une transformation $\langle m, n, 0\rangle$ telle que $\frac{n}{m} < 1 + \epsilon$. En d'autres termes, le temps réel peut être approché par du temps linéaire aussi près que voulu mais ne peut être atteint.

3.1.3 Prototypes ayant de bonnes propriétés

La définition des automates cellulaires intrinsèquement universels est abstraite et peut laisser penser que de tels automates cellulaires sont difficiles à construire. Pour se convaincre du contraire, nous présentons l'idée de la construction d'un automate cellulaire intrinsèquement universel possédant de bonnes propriétés combinatoires qui seront utilisées dans le reste du chapitre. La pésentation est délibérément informelle car l'écriture explicite de l'automate cellulaire en question n'apporte rien et obscurcirait inutilement la construction.

Un automate cellulaire intrinsèquement universel doit, par définition, être capable de reproduire, à un découpage spatial et temporel et une translation près, le comportement de tout automate cellulaire. De plus, d'après les résultats du chapitre précédent, il suffit de se restreindre aux automates cellulaires de rayon de voisinage 1. L'automate cellulaire intrinsèquement universel que nous construisons ressemble à une machine de Turing multi-têtes : l'ensemble d'états de l'automate cellulaire est de la forme $\Sigma \times (Q \cup \{\cdot\})$ où Σ est l'alphabet du « ruban » de dimension d, Q est l'ensemble d'états des têtes et $\cdot$ dénote l'absence de tête sur une cellule.

L'idée de la construction est de découper l'espace $\mathbb{Z}^d$ régulièrement en cubes de cellules. Chaque cube est une macro-cellule codant une cellule de l'automate cellulaire simulé. Pour marquer cette idée, les cellules situées à la frontière entre deux cubes sont marquées par un état spécial $\# \in \Sigma$ indiquant la frontière.

Chaque macro-cellule contient à chaque instant une unique tête, située au début de chaque transition de l'automate cellulaire simulé dans le même coin du cube et dans le même état $q_0 \in Q$. Les têtes de chaque macro-cellule évoluent en parallèle comme s'il s'agissait d'un peigne géant : lors d'une simulation si une tête se déplace dans une direction, toutes les têtes se déplacent dans cette direction, c'est leur état qui porte l'information et qui peut donc être différent.

À l'intérieur du cube, un certain nombre de zones sont découpées et identifiables par la tête. À un endroit précis est stockée la table de transition de l'automate cellulaire simulé. Dans une autre zone est stocké l'état courant de la macro-cellule. Dans autant de zones que nécessaire seront stockés les états de tous les voisins situés dans le rayon de voisinage de l'automate cellulaire simulé. Enfin, une dernière zone sert à stocker le résultat d'une transition. Tous ces codages sont effectués sur un alphabet de taille fixée.

Une transition de la cellule simulée est effectuée de la manière suivante. Tout d'abord, tous les états des cellules situées dans le rayon de voisinage sont recopiés dans les zones prévues à cet effet. Puis, la table de transition est parcourue dans son intégralité. À chaque étape, les valeurs des états stockés sont comparés à la table de transition et une copie du résultat est réalisée, ou simulée s'il n'y a pas correspondance. Une fois le parcours de la table de transition terminé, la tête recopie le résultat de la transition dans la zone de stockage de l'état de la cellule et nettoie toutes les autres aires de stockages. Puis, la tête regagne son coin et passe dans l'état q_0.

En particulier, la synchronicité de la simulation est garantie par le déplacement en peigne. L'information obtenue chez les voisins l'est à travers la frontière matérialisée par $\#$. En posant un compteur unaire le long de la frontière, une tête peut retrouver sa cellule originelle sans se perdre.

Ainsi décrit, cet automate cellulaire simule tout automate cellulaire avec pour découpage spatial celui correspondant au cube de cellules *ad hoc*, pour découpage temporel le temps de réaliser une transition et aucune translation.

3.1.4 Minimisation

La construction de très petites machines universelles est une activité ancienne, introduite par C. E. Shannon [15], et qui repose habituellement sur des codages *ad hoc* très compliqués. Cette problématique est explorée pour d'autres modèles que les machines de Turing, par exemple pour les machines de Post, voir M. Kudlek et Y. Rogozhin [78]. Dans le cas des automates cellulaires, l'universalité intrinsèque présente, dans ce cadre, l'intérêt de mettre à jour de nouvelles méthodes pour repousser le calcul de la règle locale de transition vers un niveau plus global.

Dans ses travaux, J. von Neumann [80] introduit un automate cellulaire Turing universel qui se trouve être, *a posteriori*, intrinsèquement universel. À notre connaissance, la question de chercher de petits automates cellulaires universels remonte aux travaux de A. R. Smith III [22] et E. R. Banks [20] fut le premier à l'exprimer en terme d'universalité intrinsèque. E. R. Banks a clos ce problème en dimension 2 et au-dessus. Dans le cadre des automates cellulaires de dimension 1, le problème est plus difficile car le croisement de l'information est moins aisé. Jusqu'à la construction de notre automate intrinsèquement universel à 6 états (présenté plus loin) il existait une différence de taille importante entre le plus petit automate cellulaire Turing universel et le plus petit automate cellulaire intrinsèquement universel. La table 3.1 présente certains résultats précédents. Le dernier élément du tableau concerne la présentation non formelle faite par S. Wolfram [91] qui nous semble peu convaincante à ce jour.

année	auteur	d	ν	états	universalité
1966	J. von Neumann [80]	2	5	29	intrinsèque
1968	E. F. Codd [82]	2	5	8	intrinsèque
1970	E. R. Banks [20]	**2**	**5**	**2**	**intrinsèque**
		1	3	18	intrinsèque
		1	**5**	**2**	**intrinsèque**
1971	A. R. Smith III [22]	2	7	7	Turing
		1	3	18	Turing
1982	E. Y. Berlekamp *et al.* [85]	2	8	2	intrinsèque
1987	J. Albert et K. Čulik II [34]	1	3	14	intrinsèque
1990	K. Lindgren et M. G. Nordhal [40]	**1**	**3**	**7**	**Turing**
2002	*M. Cook et S. Wolfram [91]*	*1*	*3*	*2*	*Turing ?*

où d est la dimension de l'automate cellulaire et ν la taille du voisinage : 3 et 5 correspondant aux voisinages de von Neumann en dimension 1 et 2 et 8 au voisinage de Moore en dimension 2.

TAB. 3.1 – Petits automates cellulaires universels remarquables

3.2 Indécidabilité de l'universalité intrinsèque

Nous allons démontrer notre résultat d'indécidabilité par réduction à une variante du résultat de J. Mazoyer et I. Rapaport [70], CA-1D-NIL-PER.

CA-1D-NIL-PER	
Entrée	Un automate cellulaire unidimensionnel $\mathcal{A}$ ainsi qu'un état distingué q de $\mathcal{A}$
Question	L'automate $\mathcal{A}$ est-il q-nilpotent pour les configurations périodiques ?

Certaines propriétés dynamiques des automates cellulaires ne sont ni immédiates ni asymptotiques. Un exemple d'une telle propriété est l'universalité intrinsèque. Dans le cas classique des machines de Turing, l'universalité pour le calcul, même non complètement formalisée, est indécidable par application du théorème de Rice. Dans le cas de la dynamique des automates cellulaires, il n'existe pas de tel outil. Nous allons démontrer que le problème CA-1D-UNIV est indécidable à l'aide d'une nouvelle technique.

CA-1D-UNIV	
Entrée	Un automate cellulaire unidimensionnel $\mathcal{A}$
Question	L'automate $\mathcal{A}$ est-il intrinsèquement universel ?

3.2.1 Un automate cellulaire intrinsèquement universel

Notre démonstration de l'indécidabilité de l'universalité intrinsèque pour les automates cellulaires de dimension 1 procède par réduction au problème de la nilpotence pour les configurations périodiques. De plus, cette démonstration repose sur la construction d'un automate cellulaire intrinsèquement universel $\mathcal{U}$ un peu particulier. Nous décrivons rapidement sa structure et ses propriétés.

L'automate cellulaire $\mathcal{U}$ est défini par simulation d'une machine de Turing à plusieurs têtes $\mathcal{M} = (Q_{\mathcal{U}}, \Sigma_{\mathcal{U}}, \pi)$ d'ensemble d'états $Q_{\mathcal{U}}$, d'alphabet $\Sigma_{\mathcal{U}}$ et dont la règle locale de transition π associe $Q_{\mathcal{U}} \times \Sigma_{\mathcal{U}}$ à $Q_{\mathcal{U}} \times \Sigma_{\mathcal{U}} \times \{\leftarrow, \downarrow, \rightarrow\}$.

- Notons que le comportement de $\mathcal{M}$ sur les configurations où plusieurs têtes se rencontrent est indéfini.

Par simulation, on entend que $\mathcal{U}$ est défini par

$$\mathcal{U} = (\mathbb{Z}, (\{\cdot\} \cup Q_{\mathcal{U}}) \times \Sigma_{\mathcal{U}}, N_{\mathsf{vN}}, \delta_{\mathcal{U}}) .$$

Un état de $\mathcal{U}$ est une paire constituée d'une tête ou d'un blanc et d'une lettre. Une configuration de $\mathcal{U}$ est identique à une configuration de $\mathcal{M}$·

$$\begin{array}{cccccccccccccccccccccc} a & a & a & a & b & b & a & b & a & b & a & a & a & b & b & a & a & a & b & a & b & b \\ \cdot & \cdot & \cdot & \cdot & \cdot & q & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & q' & \cdot & \cdot & \cdot & \cdot & \cdot \end{array}$$

La règle locale de transition $\delta_{\mathcal{U}}$ est définie de façon à simuler $\mathcal{M}$ selon la fonction π sur les configurations localement valides de $\mathcal{M}$. Par configurations localement valides de $\mathcal{M}$, il faut entendre les configurations telles que deux têtes Turing ne sont pas dans le voisinage de von Neumann l'une de l'autre. Pour compléter la définition de $\delta_{\mathcal{U}}$, nous requérons simplement qu'aucun tête ne soit créée. À titre d'exemple, les têtes qui invalident une configuration peuvent être effacées.

Nous imposons aussi quelques contraintes sur la simulation. Pour tout automate cellulaire $\mathcal{A}$, il existe un entier positif m et une application injective φ de $Q_{\mathcal{A}}$ dans $Q_{\mathcal{U}}^m$ telle que $\mathcal{U}$ simule $\mathcal{A}$ selon φ sans translation (il existe un entier n tel que $\overline{\varphi} \circ \mathcal{A} = \mathcal{U}^{\langle m,n,0\rangle} \circ \overline{\varphi}$) et telle que φ a les trois propriétés suivantes :

Premièrement, chaque macro-cellule est pilotée par une tête Turing. Formellement, il existe un état q_0 de $\mathcal{M}$ tel que, pour tout état q de $\mathcal{A}$, son image $\varphi(q)$ contient q_0 comme composante tête de sa première cellule et aucun autre tête ailleurs dans la macro-cellule.

Deuxièmement, au cours de la simulation, les têtes Turing se déplacent en peigne. Formellement, pour toute configuration $\mathcal{C}$ de $\mathcal{A}$ et pour tout temps t, les composantes tête de la configuration $\mathcal{C}^{(t)} = G_{\mathcal{U}}^t\left(\overline{\varphi}(\mathcal{C})\right)$ de $\mathcal{U}$ contiennent des têtes exactement en les positions $(mi + l_t)_{i\in\mathbb{Z}}$ pour une certaine valeur de l_t.

Troisièmement, les macro-cellules de la simulation peuvent êtres agrandies à de plus grandes valeurs de m par remplissage tout en conservant les mêmes propriétés. Formellement, il existe une lettre a de $\mathcal{M}$ telle que, pour tout entier positif l, la somme $m + l$ et l'application injective

$$\begin{array}{rcccl} \varphi_l & : & Q_{\mathcal{A}} & \longrightarrow & Q_{\mathcal{U}}^{m+l} \\ & & q & \longmapsto & (\varphi(s))\,(\cdot, a)^l \end{array}$$

sont des choix valides pour remplacer m et φ et conserver les mêmes propriétés.

3.2.2 Automates cellulaires chaudières

Une fois convaincu de l'existence de $\mathcal{U}$, considérez la transformation suivante qui est la clé de la réduction. Pour tout automate cellulaire $\mathcal{A}$ et tout état q de $\mathcal{A}$, on définit un automate cellulaire produit $\mathcal{A} \circledast_q \mathcal{U}$ par

$$\mathcal{A} \circledast_q \mathcal{U} = (\mathbb{Z}, Q_{\mathcal{A}} \times \{\cdot, *\} \times Q_{\mathcal{U}}, N_{\mathcal{A}} \cup N_{\mathsf{OCA}} \cup N_{\mathcal{U}}, \delta)\,,$$

un automate cellulaire à trois niveaux dont la règle locale de transition δ est définie niveau par niveau.

Le niveau inférieur est le niveau de production d'énergie. Il est constitué d'une configuration de $\mathcal{A}$ qui évolue selon la règle locale de transition $\delta_{\mathcal{A}}$ de $\mathcal{A}$.

Le niveau intermédiaire est le niveau de diffusion d'énergie. Il est constitué d'une configuration dont les états sont $\cdot$ (aucune énergie) et $*$ (une particule énergétique). L'évolution de ce niveau est celle d'une translation sauf que : si le niveau inférieur d'une cellule ne contient pas l'état q, le niveau intermédiaire produit une particule énergétique $*$; si le niveau supérieur d'une cellule contient une tête Turing et que son niveau intermédiaire reçoit une particule énergétique $*$ d'un de ses voisins, le niveau intermédiaire dissipe l'énergie entrante et passe dans l'état sans énergie.

Le niveau supérieur est le niveau de consommation de l'énergie. Il est constitué d'une configuration de $\mathcal{U}$ qui évolue selon la règle locale de transition de $\mathcal{U}$ sous le contrôle du niveau intermédiaire. Si le niveau supérieur d'une cellule identifie une tête Turing dans le niveau supérieur d'une de ses voisines, le niveau supérieur évolue selon la règle locale de transition de $\mathcal{U}$.

3.2.3 Résultat

Lemme 1. *Pour tout automate cellulaire $\mathcal{A}$, l'automate cellulaire $\mathcal{A} \circledast_q \mathcal{U}$ est intrinsèquement universel si et seulement si $\mathcal{A}$ n'est pas q-nilpotent sur les configurations périodiques.*

Démonstration. Soit $\mathcal{A}$ un automate cellulaire et q un état de $\mathcal{A}$. La démonstration se fait par discrimination sur la q-nilpotence de $\mathcal{A}$ restreint aux configurations périodiques.

Si $\mathcal{A}$ n'est pas q-nilpotent sur les configurations périodiques, alors il existe une configuration $\mathcal{C}$ de $\mathcal{A}$ périodique en temps et en espace et non q-monochromatique. Soit p une période spatiale de $\mathcal{C}$. Pour montrer que $\mathcal{A} \circledast_q \mathcal{U}$ est intrinsèquement universel, considérons un automate cellulaire $\mathcal{B}$. Soit m et φ des paramètres d'une simulation de $\mathcal{B}$ par $\mathcal{U}$ tels que p divise m (utiliser le remplissage si nécessaire). Alors, $\mathcal{A} \circledast_q \mathcal{U}$ simule $\mathcal{B}$ avec pour paramètres m et ψ où ψ est obtenue à partir de φ selon le principe suivant. Le niveau inférieur de ψ est constitué de périodes de $\mathcal{C}$. Le niveau supérieur de ψ est donné directement par φ. Comme $\mathcal{B}$ est périodique et que sa période divise m, il produit périodiquement de l'énergie pour la simulation, préserve la structure de peigne, permettant ainsi une vraie simulation avec un éventuel retard en temps. Par le principe des tiroirs, il est possible de choisir un niveau intermédiaire pour ψ tel que les deux propriétés suivantes soient vérifiées : ψ est injective et $\mathcal{A} \circledast_q \mathcal{U}$ simule $\mathcal{B}$.

Si $\mathcal{A}$ est q-nilpotent sur les configurations périodiques, nous montrons que $\mathcal{A} \circledast_q \mathcal{U}$ est incapable de simuler l'automate cellulaire $\mathcal{B} = (\mathbb{Z}, \{\circ, \bullet\}, N_{\mathsf{OCA}}, \oplus)$ où $(\{\circ, \bullet\}, \oplus)$ est le groupe cyclique $(\mathbb{Z}_2, +)$. Supposons que $\mathcal{A} \circledast_q \mathcal{U}$ soit intrinsèquement universel. En particulier, il simule $\mathcal{B}$: soit $w_\circ$ et $w_\bullet$ les codages respectifs des états. L'automate cellulaire $\mathcal{B}$ admet un diagramme espace-temps Δ périodique à la fois en espace et en temps, avec pour motif de remplissage :

$$\begin{matrix} \circ & \bullet & \bullet \\ \bullet & \circ & \bullet \\ \bullet & \bullet & \circ \end{matrix}$$

Comme $\mathcal{A}$ est q-nilpotent sur les configurations périodiques, le niveau inférieur du diagramme espace-temps de la simulation de $\mathcal{B}$ sur Δ cesse de produire de l'énergie sur le niveau intermédiaire au bout d'un temps fini. Ainsi, $w_\circ$ et $w_\bullet$ ne peuvent contenir tous les deux des particules énergétiques sur leur niveau intermédiaire et des têtes sur leur niveau supérieur (car la tête consumerait toute l'énergie disponible en un temps fini ce qui est un problème pour garantir l'injectivité de la simulation).

S'il y a une tête dans le niveau supérieur de $w_\circ$ ou $w_\bullet$, alors le niveau intermédiaire est vide de toute particule énergétique et $\mathcal{A} \circledast_q \mathcal{U}$ se comporte comme un automate cellulaire périodique sur les configurations codantes.

S'il n'y a pas de tête dans le niveau supérieur de $w_\circ$ et $w_\bullet$ alors $\mathcal{A} \circledast_q \mathcal{U}$ se comporte comme le produit cartésien d'une translation et d'un automate périodique sur les configurations codantes.

Dans les deux cas, ces comportements ne parviennent pas à simuler le comportement de $\mathcal{B}$ car le comportement de $\mathcal{B}$ sur la configuration $\circ$-monochromatique avec la cellule 0 dans l'état $\bullet$ ne peut être simulée par un tel automate cellulaire. En effet, $\mathcal{B}$ produit un triangle de Pascal modulo deux (comme représenté sur la figure 3.1) qui ne peut être obtenu comme produit d'une translation et d'un automate cellulaire périodique. ■

Théorème 16. *Le problème de l'universalité intrinsèque pour les automates cellulaires de dimension* 1 *est indécidable.*

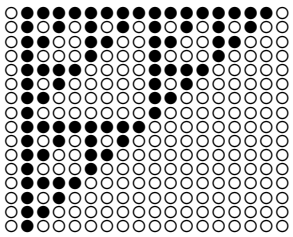


FIG. 3.1 – Triangle de Pascal modulo deux engendré par $\mathcal{B}$

Démonstration. Cette proposition est un corollaire du lemme précédent. Le calcul de $\mathcal{A} \circledast_q \mathcal{U}$ à partir de $\mathcal{A}$ étant récursif, la décidabilité du problème de l'universalité intrinsèque pour les automates cellulaires de dimension 1 impliquerait la décidabilité de la q-nilpotence sur les configurations périodiques. ■

Classiquement, les propriétés dynamiques des automates cellulaires connues pour être indécidables concernent des propriétés obtenues en un temps de calcul ou asymptotiquement. Notre démonstration de l'indécidabilité de l'universalité intrinsèque pour les automates cellulaires de dimension 1, qui est une propriété dynamique d'un genre différent, utilise une réduction originale à une propriété dynamique asymptotique.

L'idée principale de la démonstration est de combiner un automate intrinsèquement universel avec un autre automate cellulaire qui agit comme un pourvoyeur d'énergie. Le premier automate cellulaire consomme l'énergie fournie par le second. L'automate cellulaire produit est intrinsèquement universel si et seulement si le pourvoyeur d'énergie n'est pas nilpotent sur les configurations périodiques. La technique doit pouvoir s'adapter à d'autres propriétés que l'universalité intrinsèque pour démontrer de nouveaux résultats d'indécidabilité.

3.3 Un automate universel bilinéaire à deux états

Un automate cellulaire polynomial est un automate cellulaire dont l'ensemble d'états est un anneau commutatif fini et dont la règle locale de transition est exprimable comme un polynôme sur cet anneau (*i.e.* de la forme $\delta(q_1, \ldots, q_n) = \sum_{i=0}^{k} a_i q_1^{p_1^i} q_2^{p_2^i} \ldots q_n^{p_n^i}$). Un automate cellulaire bilinéaire est un automate cellulaire polynomial dont la règle locale de transition est une application bilinéaire. Bartlett et Garzon [56] ont montré qu'il existait des automates cellulaires bilinéaires universels dans le cas particulier où l'ensemble d'états est du type $\mathbb{Z}_p^p$ avec p premier, en exhibant un automate cellulaire intrinsèquement universel de ce type à 211^{211} états ! Nous montrons que même dans le cas le plus simple, où l'ensemble d'états est $\mathbb{Z}_2$, il existe des automates cellulaires intrinsèquement universels.

Définition 7. *Un* automate cellulaire bilinéaire *est un automate cellulaire quadratique dont le polynôme n'est constitué que de monômes quadratiques :*

$$\delta\left(q_1, \ldots, q_d\right) = \sum_{i=1}^{d} \sum_{j=1}^{d} b_{i,j} q_i q_j \quad .$$

Théorème 17. *Tout automate cellulaire unidirectionnel peut être simulé par un automate cellulaire bilinéaire à deux états.*

Démonstration. Soit $\mathcal{A}$ un automate cellulaire unidirectionnel $(Q, \{-1, 0\}, \delta)$. Soit n la cardinalité de Q. À un renommage des états près, on peut considérer que Q est l'ensemble $[\![0, n-1]\!]$. Construisons un automate cellulaire bilinéaire à deux états $(\mathbb{Z}_2, \{-r, \ldots, r\}, P)$ qui simule $\mathcal{A}$.

L'idée première de cette construction est de représenter une cellule d'une configuration de $\mathcal{A}$ par un bloc de cellules toutes sauf une dans l'état 0. La position de la cellule d'état 1 détermine donc l'état de la cellule de la configuration de $\mathcal{A}$. Pour coder la transition $\delta(i, j) = k$, on construit un monôme $q_s q_t$ où s et t sont les distances entre les cellules dans l'état 1 dans les codages de i, j et k comme représenté sur la figure 3.2. Pour éviter d'avoir à multiplier ce monôme par $(1 - q_l)$ pour tout l entre s et t, il faut que le codage garantisse que toutes les cellules concernées sont nécessairement dans l'état 0. De plus, pour éviter toute collision entre monômes, il faut s'assurer que l'application qui à (i, j, k) associe (s, t) est injective.

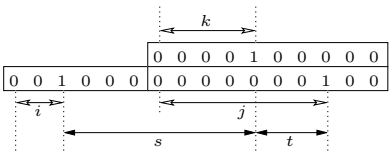


FIG. 3.2 – Principe de codage d'une cellule

Pour construire un tel codage valide, nous commençons par construire des briques de base. Les blocs d'états sont construits comme des mots sur l'alphabet $\mathbb{Z}_2$. L'opération $\cdot$ est l'opération de concaténation des mots. La fonction idx associe à un mot contenant une unique lettre 1 le nombre de lettre 0 qui précédent cette lettre 1 (par exemple $\mathrm{idx}(000100) = 3$).

Valeurs. Elles codent simplement des éléments de $\mathbb{Z}_2$. Ce sont des mots de deux lettres finissant toujours par 0.

$$v(a) = a \cdot 0$$

Blocs vides. Ils codent un bloc de n fois la valeur de 0.

$$\begin{aligned} z_n &= v(0)^n \\ |z_n| &= 2n \end{aligned}$$

Blocs codants. Ils codent une suite de valeurs contenant une unique fois la valeur 1 à une position donnée.

$$\begin{aligned} b_n(i) &= v(0)^i \cdot v(1) \cdot v(0)^{n-i-1} \\ |b_n(i)| &= 2n \\ \mathrm{idx}(b_n(i)) &= 2i \end{aligned}$$

Enveloppe de cellule. C'est l'élément fondamental de la construction. Une enveloppe de cellule (i, j) est simplement une suite de n blocs tous égaux à z_n sauf le i-ème bloc qui est égal à $b_n(j)$. La figure 3.3 représente une telle enveloppe.

$$\begin{aligned} c_n(i,j) &= z_n^i \cdot b_n(j) \cdot z_n^{n-i-1} \\ |c_n(i,j)| &= 2n^2 \\ \mathrm{idx}(c_n(i,j)) &= 2ni + 2j. \end{aligned}$$

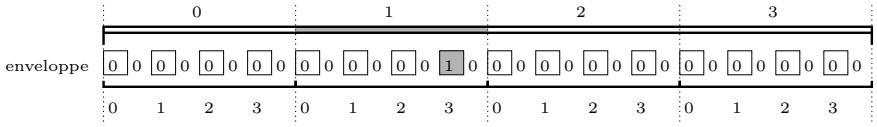


FIG. 3.3 – Une enveloppe de cellule $(1, 3)$

Pour assurer l'injectivité du codage, on tient compte de la parité de la position de la cellule à coder dans la configuration de $\mathcal{A}$. De plus, la taille des enveloppes de cellule est contrôlée par un paramètre α qui sera fixé par la suite.

Cellules paires. Elles sont représentées par une enveloppe de cellule (i, i) de taille αn lorsqu'elles codent l'état i. Une cellule paire est de longueur paire.

$$\begin{aligned} e_n(i) &= c_{\alpha n}(i,i) \\ |e_n(i)| &= 2\alpha^2 n^2 \\ \mathrm{idx}(e_n(i)) &= 2\alpha n i + 2i \end{aligned}$$

Cellules impaires. Elles sont représentées par une enveloppe de cellule $(i, 0)$ de taille αn à laquelle on concatène la lettre 0 lorsqu'elles codent l'état i. Une cellule impaire est de longueur impaire.

$$\begin{aligned} o_n(i) &= c_{\alpha n}(i,0) \cdot 0 \\ |o_n(i)| &= 2\alpha^2 n^2 + 1 \\ \mathrm{idx}(o_n(i)) &= 2\alpha n i \end{aligned}$$

Il faut maintenant montrer que cette construction est compatible avec les hypothèses faites en début de démonstration. Il y a deux propriétés à vérifier.

Pour commencer, montrons que α peut être choisi de sorte que, étant donné les positions de deux lettres 1 dans une configuration codante, on peut décider si elles correspondent à deux cellules voisines. La distance maximale entre deux lettres 1 de deux cellules voisines est

$$|o_n(0)| + \mathrm{idx}(e_n(n)) - \mathrm{idx}(o_n(0)) = \left(2\alpha^2 + 2\alpha\right) n^2 + 2n + 1 \quad .$$

La distance minimale entre deux lettres 1 de deux cellules non voisines est

$$|e_n(n)| + |o_n(i)| + \mathrm{idx}(e_n(0)) - \mathrm{idx}(e_n(n)) = \left(4\alpha^2 - 2\alpha\right) n^2 - 2n + 1 \quad .$$

Donc α doit satisfaire la contrainte

$$\left(4\alpha^2 - 2\alpha\right) n^2 - 2n + 1 > \left(2\alpha^2 + 2\alpha\right) n^2 + 2n + 1$$

qui se simplifie en

$$\alpha^2 - 2\alpha > \frac{2}{n}.$$

Un choix possible est $\alpha = 3$. La première propriété est vérifiée.

Il faut maintenant vérifier que pour tout monôme $q_s q_t$ codant une transition $\delta(i,j) = k$, l'application qui à (i,j,k) associe (s,t) est injective. Calculons les valeurs de s et t en fonction de i, j et k. Il y a deux cas, en fonction de la parité des cellules concernées. Si i correspond à une cellule paire, s est égal à

$$-\operatorname{idx}(o_n(k)) - |e_n(i)| + \operatorname{idx}(e_n(i)) = 2\alpha n(i-k) + 2i - 2\alpha^2 n^2$$

et t est égal à

$$-\operatorname{idx}(o_n(k)) + \operatorname{idx}(o_n(j)) = 2\alpha n(j-k).$$

Si i correspond à une cellule impaire, s est égal à

$$-\operatorname{idx}(e_n(k)) - |o_n(i)| + \operatorname{idx}(o_n(i)) = (2\alpha n + 2)(i-k) - 2i - 2\alpha^2 n^2 - 1$$

et t est égal à

$$-\operatorname{idx}(e_n(k)) + \operatorname{idx}(e_n(j)) = (2\alpha n + 2)(j-k).$$

Considérons un monôme $q_s q_t$ avec $s < t$. On commence par regarder la parité de s : c'est la parité de la cellule correspondant à i. Puis, on observe s modulo n : si la cellule est paire cette quantité est égale à $2i$ sinon elle est égale à $-2k-1$. De cela on déduit soit la valeur de i soit la valeur de k. Puis, de s on déduit les valeurs de i et de k car on connaît n et α. De t, connaissant k, on trouve la valeur de j. Ce qui achève la vérification de la deuxième propriété. On peut alors écrire le polynôme P :

$$P(q_{-r}, \ldots, q_r) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \begin{array}{l} q_{2\alpha n(i-\delta(i,j))+2i-2\alpha^2 n^2} q_{2\alpha n(j-\delta(i,j))} \\ + q_{(2\alpha n+2)(i-\delta(i,j))-2i-2\alpha^2 n^2-1} q_{(2\alpha n+2)(j-\delta(i,j))} \end{array}$$

Le rayon r peut être choisi égal à $4\alpha^2 n^2 + 1$. ■

Corollaire 5. *Il existe des automates cellulaires bilinéaires intrinsèquement universels à deux états.*

3.4 Un automate universel à 6 états

3.4.1 Un automate cellulaire universel simple à 8 états

Nous commençons par décrire un automate cellulaire universel à 8 états. La construction est mise en oeuvre en deux étapes. On commence par décrire le fonctionnement macroscopique de l'automate avant d'entrer dans les détails du codage microscopique de ces idées.

Considérations macroscopiques

Pour simuler le comportement d'un automate cellulaire, il faut itérer l'évaluation en parallèle d'une règle locale de transition $\delta : Q^N \to Q$. Soit Ξ l'alphabet binaire $\{0, 1\}$ et $n = \lceil \log_2 |Q| \rceil$. On peut encoder les éléments de Q par des éléments distincts de Ξ^n et ainsi décomposer δ en n fonctions booléennes $\delta_i : \Xi^{Nn} \to \Xi$. La figure 3.4 illustre une telle décomposition sur un automate cellulaire qui servira d'exemple tout au long de cette construction. L'opérateur booléen binaire `NAND`, noté $|$, formant à lui seul une base complète[1] pour le calcul booléen, on peut considérer des représentations des fonctions (δ_i) par des circuits booléens dont les seuls portes sont des portes `NAND`. Aussi, pour simuler le comportement d'un automate cellulaire, il suffit d'évaluer n circuits booléens en parallèle en une même durée.

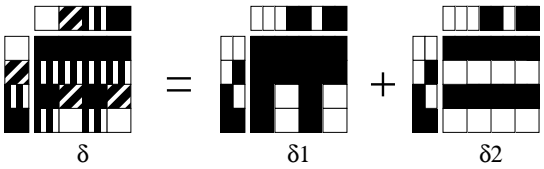


FIG. 3.4 – Règle locale d'un automate cellulaire et sa décomposition

Plutôt que d'utiliser des circuits booléens classiques, nous choisirons des circuits à niveaux avec deux types de portes : copie (la fonction booléenne identité) et `NAND` (la fonction booléenne NON-ET). Dans un circuit à niveaux, les portes sont réparties en un nombre fini de niveaux L_0, L_1, ..., L_k, de telle sorte que les variables apparaissent au niveau L_0, la sortie au niveau L_k et qu'une porte située au niveau L_{i+1} prenne ses entrées exclusivement au niveau L_i. Une telle représentation de fonctions booléennes est présentée sur la figure 3.5. Afin de simuler des automates cellulaires, nous voulons simuler le calcul à l'aide de tels circuits. Il faut tout d'abord aplatir les circuits pour les encoder dans un $Q^{\mathbb{Z}}$.

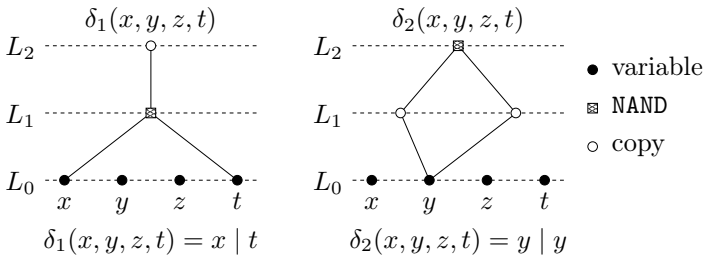


FIG. 3.5 – Fonctions booléennes et représentation par des circuits à niveaux

Définition 8. *Un simulateur booléen d'automates cellulaires est un modèle dynamique discret. Dans ce modèle, chaque cellule stocke une valeur booléenne qui est mise à jour à chaque temps à l'aide de certaines informations constantes de la cellule : un opérateur (copie ou `NAND`) et la position relative des opérandes dans la ligne de cellules.*

[1] Une base complète B pour le calcul booléen est un ensemble de fonctions booléennes tel que toute fonction booléenne puisse être représentée par un circuit dont les portes calculent des fonctions de B. On trouvera une introduction aux fonctions booléennes dans [87].

Les opérandes d'un opérateur pouvant se situer arbitrairement éloignées, un simulateur booléen d'automates cellulaires n'est pas un automate cellulaire. Cependant, il peut simuler n'importe quel automate cellulaire. Partant de la représentation par des circuits à niveaux des fonctions δ_i associées à la règle locale de transition de l'automate cellulaire, on ajoute si nécessaire des niveaux de copie aux circuits afin qu'ils aient tous le même nombre de niveaux $T + 1$. Une cellule de l'automate cellulaire est alors encodée par une macro-cellule : un bloc de cellules consistant en la concaténation des circuits des précédents. Les variables sont extraites des sorties des autres macro-cellules. En T étapes de temps, chaque macro-cellule effectue exactement une transition de l'automate cellulaire simulé. La figure 3.6 présente un tel codage.

		δ_1		δ_2			δ_1		δ_2			δ_1		δ_2			
valeur booléenne	···	?	x	?	?	y	?	z	?	?	t	?	u	?	?	v	···
opérateur		|	C	C	C	|	|	C	C	C	|	|	C	C	C	|	
opérande 1		1	1	8	9	1	1	1	8	9	1	1	1	8	9	1	
opérande 2		9				2	9				2	9				2	
				q_0					q_1					q_2			

Dans cet exemple, une macro-cellule q_j encode une cellule de l'automate cellulaire. Seules les valeurs booléennes codantes sont données, `?` correspond à du bruit. Pour le codage des opérateurs, `|` correspond à `NAND` et `C` à copie. Un pas de simulation est effectué en 2 temps du simulateur (un temps par niveau).

FIG. 3.6 – Codage de cellules dans le simulateur booléen d'automates cellulaires

Pour simuler un automate cellulaire donné, le simulateur n'utilise qu'un nombre fini de cellules distinctes. Ainsi, les opérandes accédées sont à une distance maximale m (une bonne borne pour m dans notre construction est n fois la taille du plus gros circuit à niveaux pour une fonction booléenne à Nn entrées). Pour simuler le fonctionnement du simulateur booléen d'automates cellulaires par un automate cellulaire, on propose de coder une cellule du simulateur par approximativement m cellules de l'automate cellulaire. Chaque cellule du simulateur enverra sa valeur booléenne à chacune de ses m voisines sur la droite. Lorsque la i-ème valeur rencontre la macro-cellule, celle-ci compulse la i-ème cellule de son bloc et, en fonction de son contenu, copie la valeur, applique un opérateur ou ignore la valeur. Lorsque m valeurs ont traversé la macro-cellule, la macro-cellule dispose de sa nouvelle valeur booléenne et peut alors la transmettre à ses voisines.

Codage microscopique

Support de fond. Chaque cellule du simulateur est représentée par $m + 2$ cellules de notre automate universel, où m majore la distance maximale des opérandes. Les états sont divisés en trois. La première sorte, l'état Op, concerne m cellules et est utilisée pour décider des opérations à effectuer, comme décrit plus haut. Une cellule est dans un état de la seconde sorte Val dont la fonction est de mémoriser la valeur booléenne courante de la macro-cellule. Enfin, une dernière cellule est dans un état de troisième sorte Sig qui transmet une valeur booléenne de macro-cellule en macro-cellule. Un calcul a lieu à

chaque rencontre des trois sortes d'états. Une vue détaillée des interactions est représentée sur la figure 3.7.

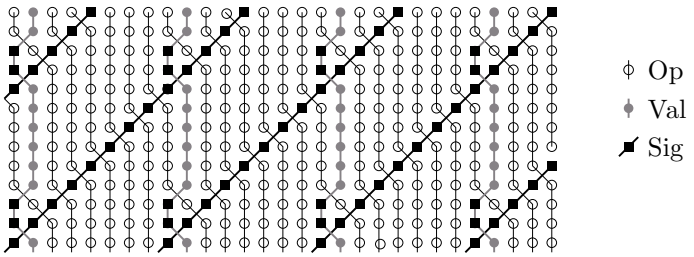


FIG. 3.7 – Trois types d'états pour déplacer l'information

Sur la figure, le temps va de bas en haut. Les états Val sont ceux qui restent sur une même colonne. Les états Sig se déplacent en temps réel vers la droite. Les états Op se déplacent lentement vers la gauche, à chaque croisement d'un état Sig. La précédente figure définit (partiellement) un automate cellulaire à trois états et voisinage des premiers voisins. La règle locale de transition de cet automate est donnée par la table 3.2.

q_l	q_m	q_r	$\delta(q_l, q_m, q_r)$	q_l	q_m	q_r	$\delta(q_l, q_m, q_r)$
Op	Op	Op	Op	Sig	Op	Op	Sig
Op	Op	Sig	Op	Sig	Op	Sig	Val
Op	Op	Val	Op	Sig	Op	Val	Sig
Op	Sig	Op	Op	Sig	Sig	Op	Op
Op	Sig	Sig	Sig	Sig	Val	Op	Sig
Op	Sig	Val	Sig	Val	Op	Op	Op
Op	Val	Op	Val	Val	Op	Sig	Op

TAB. 3.2 – Règle locale de transition de l'automate à 3 états

Support de calcul. Afin d'achever la construction, nous spécialisons les trois types d'états en sous-types. Les états Sig codent deux valeurs booléennes possibles Sig_0 et Sig_1. De même, les états Val sont séparés en Val_0 et Val_1. Quant aux états Op, ils sont séparés en $\mathrm{Op}_\sharp$, Op_C, Op_F et $\mathrm{Op}_|$ représentant respectivement la fin d'une macro-cellule, la copie du signal entrant, l'ignorance de ce signal et l'application de l'opérateur `NAND` entre le signal entrant et la valeur courante de la macro-cellule. On obtient ainsi un automate cellulaire à 8 états dont la règle locale de transition est décrite par la table 3.3. Cet automate est intrinsèquement universel.

Codage effectif. Le codage exact d'une cellule de l'automate cellulaire à simuler, en partant d'une macro-cellule du simulateur booléen d'automate cellulaire, est le suivant :

Soit m la plus grande position relative d'un opérande utilisée par les opérateurs de la macro-cellule (par exemple $m = 9$ sur la figure 3.6). L'enveloppe de codage des cellules est

$$\omega(x, d_1, \ldots, d_{m-1}) = \mathrm{Val}_x \, \mathrm{Op}_{d_1} \cdots \mathrm{Op}_{d_{m-4}} \, \mathrm{Sig}_x \, \mathrm{Op}_{d_{m-3}} \, \mathrm{Op}_{d_{m-2}} \, \mathrm{Op}_{d_{m-1}} \, \mathrm{Op}_\sharp \, .$$

q_l	q_m	q_r	$\delta(q_l, q_m, q_r)$	q_l	q_m	q_r	$\delta(q_l, q_m, q_r)$
Op_i	Op_j	Op_k	Op_j	Sig_v	Op_i	Op_j	Sig_v
Op_i	Op_j	Sig_v	Op_j	Sig_v	Op_i	$\mathrm{Sig}_{v'}$	$\mathrm{Val}_{\varphi(v,i,v')}$
Op_i	Op_j	Val_v	Op_j	Sig_v	Op_i	$\mathrm{Val}_{v'}$	Sig_v
Op_i	Sig_v	Op_j	Op_j	Sig_v	$\mathrm{Sig}_{v'}$	Op_i	Op_i
Op_i	Sig_v	$\mathrm{Sig}_{v'}$	Sig_v	Sig_v	$\mathrm{Val}_{v'}$	Op_i	$\mathrm{Sig}_{\psi(v,v',i)}$
Op_j	Sig_v	$\mathrm{Val}_{v'}$	$\mathrm{Sig}_{v'}$	Val_v	Op_i	Op_j	Op_i
Op_i	Val_v	Op_j	Val_v	Val_v	Op_i	$\mathrm{Sig}_{v'}$	Op_i

$$\text{où} \quad \varphi(v,i,v') = \begin{cases} v' & \text{si } i = C, \\ v \mid v' & \text{si } i = |, \\ v & \text{sinon,} \end{cases} \qquad \text{et} \quad \psi(v,v',i) = \begin{cases} v' & \text{si } i = \sharp, \\ v & \text{sinon.} \end{cases}$$

TAB. 3.3 – Règle locale de transition de l'automate universel à 8 états

Les cellules sont codées comme des instanciations, par des opérateurs, de cette enveloppe :

cellule copie $\mathtt{c}_{\mathtt{1}}$, $\quad \omega^{\mathtt{c}}_{\mathtt{1}}(x) = \omega(x, F, \ldots, F)$

cellule copie $\mathtt{c}_{\mathtt{k}}$ où $k > 1$, $\quad \omega^{\mathtt{c}}_{\mathtt{k}}(x) = \omega(x, \underbrace{F, \ldots, F}_{k-2}, C, F, \ldots, F)$

cellule $\mathtt{NAND}\ \mathtt{|}_{\substack{\mathtt{1}\\ \mathtt{k}}}$, $\quad \omega^{\mathtt{|}}_{\substack{\mathtt{1}\\ \mathtt{k}}}(x) = \omega(x, \underbrace{F, \ldots, F}_{k-2}, |, F, \ldots, F)$

cellule $\mathtt{NAND}\ \mathtt{|}_{\substack{\mathtt{k}\\ \mathtt{l}}}$ où $k > 1$, $\quad \omega^{\mathtt{|}}_{\substack{\mathtt{k}\\ \mathtt{l}}}(x) = \omega(x, \underbrace{F, \ldots, F}_{k-2}, C, \underbrace{F, \ldots, F}_{l-k-1}, |, F, \ldots, F)$

Soit ω_1, …, ω_n la suite des codages de cellules correspondant à la macro-cellule à simuler. Soit v_1,…,v_n les valeurs booléennes initiales à entrer dans le circuit pour coder un état particulier. La macro-cellule codante de l'automate cellulaire à 8 états est alors, le décalage servant à la synchronisation :

$$\omega_2(v_1)\omega_3(v_2)\cdots\omega_n(v_{n-1})\omega_1(v_n).$$

Un pas de simulation est alors obtenu en $m(m+2)T$ pas de calcul où T est le nombre de niveaux du circuit. En appliquant cette transformation à la configuration de la figure 3.6 on obtient la configuration de la figure 3.8.

```
(0,0)  1-----%---#0-----/-+-#0-----/--+#0|----/---#0-----/--|#
(0,1)  1-----%---#0-----/-+-#0-----/--+#0|----/---#1-----%--|#
(1,0)  1-----%---#1-----%-+-#0-----/--+#0|----/---#0-----/--|#
(1,1)  1-----%---#1-----%-+-#0-----/--+#0|----/---#1-----%--|#
```

Les symboles `-`, `+`, `#` et `|` représentent respectivement les operateurs Op_F, Op_C, $\mathrm{Op}_\sharp$ et $\mathrm{Op}_|$. Les symboles `0` et `1` représentent les valeurs Val_0 et Val_1. Les symboles `/` et `%` représentent les signaux Sig_0 et Sig_1. Un pas de la simulation est obtenu en 198 pas de calcul de l'automate.

FIG. 3.8 – Codage d'un automate cellulaire dans l'automate universel à 8 états

3.4.2 Affinement du nombre d'états requis

Le précédent paragraphe a présenté la construction d'un automate cellulaire intrinsèquement universel à 8 états. Par une analyse fine du précédent automate, nous montrons que seuls 7 états parmi les 8 sont nécessaires pour atteindre l'universalité. Puis, nous présentons brièvement une technique pour ramener ce nombre d'états à 6 en utilisant seulement deux types d'états Op.

Émuler la copie : de 8 à 7 états

L'opérateur booléen `NAND` se comporte de la manière suivante avec des constantes : $x \mid 0 = 1$ et $x \mid 1 = \neg x$. Dans la précédente simulation, remplaçons chaque signal Sig_x par un triplet de signaux Sig_1, Sig_x, Sig_0. L'opérateur de copie peut alors être émulé grâce à l'égalité suivante :

$$1 \mid (x \mid (0 \mid y)) = x \ .$$

Les états Op_C sont alors remplacés par la suite $\mathrm{Op}_|$, $\mathrm{Op}_|$, $\mathrm{Op}_|$ et les trois autres sous-types de Op sont conservés tels quels mais gardés par des états Op_F à gauche et à droite pour ignorer les signaux constants. Cette transformation, appliquée à la figure 3.8, permet d'obtenir la figure 3.9. Lorsqu'on applique cette transformation, le temps nécessaire à simuler une transition passe de $m(m+2)T$ à $m(3m+4)T$ pas de calcul.

```
(0,0) 1---------------%----%----/--#-   (0,1) 1---------------%----%----/--#-
      0---------------%---|/||--/--#-         0---------------%---|/||--/--#-
      0---------------%----/--||/|-#-         0---------------%----/--||/|-#-
      0-|-------------%----/----/--#-         0-|-------------%----/----/--#-
      0---------------%----/---|/--#-         1---------------%----%---|/--#-
(1,0) 1---------------%----%----/--#-   (1,1) 1---------------%----%----/--#-
      1---------------%---|%||--/--#-         1---------------%---|%||--/--#-
      0---------------%----/--||/|-#-         0---------------%----/--||/|-#-
      0-|-------------%----/----/--#-         0-|-------------%----/----/--#-
      0---------------%----/---|/--#-         1---------------%----%---|/--#-
```

Les symboles `-`, `#` et `|` représentent respectivement les opérateurs Op_F, $\mathrm{Op}_\sharp$ et $\mathrm{Op}_|$. Les symboles `0` et `1` représentent les valeurs Val_0 et Val_1. Les symboles `/` et `%` représentent les signaux Sig_0 et Sig_1. Un pas de simulation est obtenu en 558 pas de calcul de l'automate.

FIG. 3.9 – Codage d'un automate cellulaire dans l'automate universel à 7 états

On a donc obtenu un automate cellulaire intrinsèquement universel à 7 états en supprimant l'état Op_C du précédent automate universel à 8 états.

Séparer les opérateurs : de 7 à 6 états

Pour obtenir un automate cellulaire intrinsèquement universel à 6 états, nous allons maintenant modifier l'automate précédent afin d'utiliser seulement 2 sous-types d'état Op.

La figure 3.10 synthétise l'idée de cette transformation. Dans le cas de l'automate à 7 états, les seules véritables modifications d'états proviennent des opérateurs $\mathrm{Op}_\sharp$ et $\mathrm{Op}_|$ et résultent de la lecture de l'état Op à droite d'une collision entre Val et Sig. Dorénavant nous séparons le comportement de Op en un comportement Op gauche et Op droit. Les deux sous-types d'états Op sont Op_0 et Op_1. L'état Op gauche contrôle la modification de l'état Val : s'il est égal à Op_1, l'opérateur `NAND` est appliqué entre la valeur de l'état Sig et la valeur de l'état Val et le résultat est stocké dans l'état Val ; s'il est égal à Op_0, la

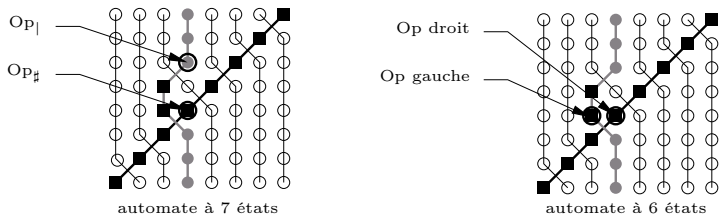


FIG. 3.10 – Lieux du calcul dans les automates cellulaires universels

valeur de l'état Val est conservée. L'état Op droit contrôle la modification de l'état Sig : s'il est égal à Op_1, la valeur de l'état Val est stockée dans l'état Sig ; s'il est égal à Op_0, la valeur de l'état Sig est conservée. L'émulation des trois anciens état Op est détaillée dans la table 3.4.

ancien Op	Op gauche	Op droit
Op_F	Op_0	Op_0
$\mathrm{Op}_\sharp$	Op_0	Op_1
$\mathrm{Op}_\mid$	Op_1	Op_0

TAB. 3.4 – Codage des états Op de l'automate à 7 états dans l'automate à 6 états

Pour rendre effective cette transformation, il faut que les états Val se déplacent dans les états Op de deux en deux. Pour permettre cela, on ajoute un nouvel état Sig entre chaque paire d'états Sig consécutifs. Le but de ce signal poubelle n'est pas de transporter de l'information utile mais simplement de décaler les états Op par rapport à l'état Val.

Cette stratégie n'est pas suffisante. En effet, il faut s'assurer que ces signaux poubelle ne perturbent pas les calculs. D'autre part, afin de préserver l'injectivité de la simulation, il faut nettoyer les signaux poubelles à la fin de chaque macro-transition.

$$\mathtt{F}_{\mathtt{F}}\mathtt{F}_{\mathtt{F}}\mathtt{F}_{\mathtt{F}}\ldots\mathtt{F}_{\mathtt{F}}\mathtt{F}_{\mathtt{F}}\mathtt{F}_{\#}\mid_{\#}\mid_{\#}\mid_{\mathtt{F}}\mathtt{F}_{\mathtt{F}}\mathtt{F}_{\mathtt{F}}\mathtt{F}_{\mathtt{F}}\ldots\mathtt{F}_{\mathtt{F}}\mathtt{F}_{\mathtt{F}}\mathtt{F}_{\mathtt{F}}\left(\mathtt{F}_{\#}\mid_{\mathtt{F}}\mathtt{F}_{\mathtt{F}}\right)\mathtt{F}_{\mathtt{F}}\mathtt{F}_{\mathtt{F}}\mathtt{F}_{\mathtt{F}}\ldots\mathtt{F}_{\mathtt{F}}\#_{\mid}\mathtt{F}_{\mathtt{F}}$$

FIG. 3.11 – Codage d'une cellule, indicé par les opérations des signaux poubelle

Perturbation du calcul. Considérons tout d'abord le problème de perturbation du calcul. La figure 3.11 présente une suite typique d'états Op rencontrés dans le codage avec l'automate à 7 états. En indice sont représentées les opérations effectuées par les signaux poubelle entre les signaux utiles. La seule opération qui modifie la valeur de l'état Val est l'application de `NAND` qui n'est effectuée par les signaux poubelle qu'à la fin du calcul. Cette opération compromet l'information. Cependant, à l'instant précédent, la valeur de l'état Val a été sauvegardée sur le signal Sig. En fait, après l'exécution de cet opération $\mathrm{Op}_\sharp$, la prochaine configuration de départ est entièrement décrite par la suite des signaux utiles. Il suffit de tout remettre en place au moment du nettoyage des signaux poubelle.

Nettoyage des signaux poubelle. L'opération de nettoyage des signaux poubelle est certainement la partie plus technique de la construction. L'astuce réside dans la configuration partielle donnée sur la figure 3.12. Après la traversée de cette configuration, les

signaux poubelle des gardes Sig_0 et Sig_1 contiennent la valeur Sig_1, le signal poubelle de Sig_x contient $\mathrm{Sig}_{\bar{x}}$ et la valeur de la macro-cellule stockée à sa gauche vaut Val_x. Si on ajoute la résolution de quelques petits problèmes de synchronisation supplémentaires, l'algorithme de codage, en partant de la configuration codée pour l'automate à 6 états, procède comme suit.

signal			0		x		1		0		y		1
Op *principal*			|		F		F		|		|		|
Op *poubelle*	#			F		F		#		#		#		F
valeur	?			1		1		1		1		$\bar{y}$		y

FIG. 3.12 – Opérations de nettoyage des signaux poubelle

Tout d'abord, pour chaque sous-mot codant une cellule du simulateur booléen d'automates cellulaires, constitué de $3m$ cellules de type Op, concaténer $n-2$ fois la configuration $\mathrm{Op}_F\,\mathrm{Op}_F\,\mathrm{Op}_F$ où n est la longueur de la macro-cellule du simulateur booléen d'automates cellulaires. Puis, y concaténer le motif de nettoyage des signaux poubelle de la figure 3.12, c'est-à-dire $\mathrm{Op}_|\,\mathrm{Op}_F\,\mathrm{Op}_F\,\mathrm{Op}_|\,\mathrm{Op}_|\,\mathrm{Op}_|$. Enfin, appliquer la conversion de 7 à 6 états décrite dans la table 3.4. Cela laissera suffisamment de place entre les signaux pour insérer les signaux poubelle en laissant une distance au moins 4 entre chaque signal (pour éviter des collisions dans le calcul). Le temps de simulation d'une transition passe de $m(3m+4)T$ à $(m+n)(6(m+n)+7)T$ pas de calcul.

```
(0,0) -1-----------------------------------------------+--------%----%----/----%+---%--+-/+-+-
      -0-----------------------------------+-+-+----------+--------%----%----%----/+---%--+-/+-+-
      -0-----------------------------------------+-+-+----+--------%----%----%----/+---%--+-/+-+-
      -0-+-----------------------------------------------+--------%----%----%----/+---%--+-/+-+-
      -0-----------------------------------------+------+--------%----%----%----/+---%--+-/+-+-
(0,1) -1-----------------------------------------------+--------%----%----/----%+---%--+-/+-+-
      -0-----------------------------------+-+-+----------+--------%----%----%----/+---%--+-/+-+-
      -0-----------------------------------------+-+-+----+--------%----%----%----/+---%--+-/+-+-
      -0-+-----------------------------------------------+--------%----%----%----/+---%--+-/+-+-
      -1-----------------------------------------+------+--------%----%----/----%+---%--+-/+-+-
(1,0) -1-----------------------------------------------+--------%----%----/----%+---%--+-/+-+-
      -1-----------------------------------+-+-+----------+--------%----%----/----%+---%--+-/+-+-
      -0-----------------------------------------+-+-+----+--------%----%----%----/+---%--+-/+-+-
      -0-+-----------------------------------------------+--------%----%----%----/+---%--+-/+-+-
      -0-----------------------------------------+------+--------%----%----%----/+---%--+-/+-+-
(1,1) -1-----------------------------------------------+--------%----%----/----%+---%--+-/+-+-
      -1-----------------------------------+-+-+----------+--------%----%----/----%+---%--+-/+-+-
      -0-----------------------------------------+-+-+----+--------%----%----%----/+---%--+-/+-+-
      -0-+-----------------------------------------------+--------%----%----%----/+---%--+-/+-+-
      -1-----------------------------------------+------+--------%----%----/----%+---%--+-/+-+-
```

Les symboles `-` et `+` représentent respectivement les opérateurs Op_0 et Op_1. Les symboles `0` et `1` représentent les valeurs Val_0 et Val_1. Les symboles `/` et `%` représentent les signaux Sig_0 et Sig_1. Un pas de la simulation est obtenu en 2548 pas de calcul de l'automate.

FIG. 3.13 – Codage d'un automate cellulaire dans l'automate universel à 6 états

Ainsi s'achève notre construction d'un automate cellulaire intrinsèquement universel à 6 états. Comme toutes les constructions algorithmiques, celle-ci est pénible à suivre mais elle tente de rendre les idées qui amènent à cet automate. Pour une démonstration purement formelle de l'universalité de cet automate, il serait suffisant de montrer que cet automate simule un automate cellulaire intrinsèquement universel connu. La règle locale de transition de notre automate est donnée par la table 3.5.

Des informations complémentaires relatives à la mise en oeuvre des constructions précédentes sont disponibles sur la page web de l'auteur, à l'adresse suivante :

q_l	q_m	q_r	$\delta(q_l, q_m, q_r)$	q_l	q_m	q_r	$\delta(q_l, q_m, q_r)$
Op_i	Op_j	Op_k	Op_j	Sig_v	Op_i	Op_j	Sig_v
Op_i	Op_j	Sig_v	Op_j	Sig_v	Op_i	$\mathrm{Sig}_{v'}$	Val_v
Op_i	Op_j	Val_v	Op_j	Sig_v	Op_i	$\mathrm{Val}_{v'}$	Sig_v
Op_i	Sig_v	Op_j	Op_j	Sig_v	$\mathrm{Sig}_{v'}$	Op_i	Op_i
Op_i	Sig_v	$\mathrm{Sig}_{v'}$	Sig_v	Sig_v	$\mathrm{Val}_{v'}$	Op_i	$\mathrm{Sig}_{\psi(v,v',i)}$
Op_j	Sig_v	$\mathrm{Val}_{v'}$	$\mathrm{Sig}_{\varphi(j,v,v')}$	Val_v	Op_i	Op_j	Op_i
Op_i	Val_v	Op_j	Val_v	Val_v	Op_i	$\mathrm{Sig}_{v'}$	Op_i

$$\text{où} \quad \varphi(j,v,v') = \begin{cases} v' & \text{si } j = 0, \\ v \mid v' & \text{si } j = 1, \end{cases} \qquad \text{et} \quad \psi(v,v',i) = \begin{cases} v & \text{si } i = 0, \\ v' & \text{si } i = 1. \end{cases}$$

TAB. 3.5 – Règle locale de transition de l'automate universel à 6 états

`http://www.ens-lyon.fr/~nollinge/6st/`.

En particulier, on y trouvera des outils de simulation et des exemples de diagrammes espace-temps des 3 automates cellulaires à 8, 7 puis 6 états construits ci-dessus.

CHAPITRE

4 POUR ALLER PLUS LOIN

Nous avons présenté, à la fin de l'introduction, les grandes directions et questions qui nous semblent pertinentes pour poursuivre l'exploration des liens entre propriétés locales et globales dans le cadre des automates cellulaires. Nous présentons ici certaines questions plus techniques, concernant des points précis de la thèse et qui nous semblent mériter un approfondissement :

- En introduisant les transformations géométriques **PCS**, nous avons défini les transformations de découpage spatial $\mathsf{P}_{F,v}$ à l'aide de réseaux de points v et de motifs finis F liés aux classes d'équivalence engendrées par les réseaux v. La théorie mathématique des réseaux de points, en particulier des réseaux de $\mathbb{Z}^d$, a été étudiée en détail, voir par exemple les travaux de J. H. Conway et N. J. A. Sloane [89]. De nombreux résultats de cette théorie exhibent des différences de structure lorsque la dimension varie. Quels liens existe-t-il entre les propriétés des réseaux de points sous-jacents et les propriétés structurelles des automates cellulaires de dimensions différentes ?
- Le groupage est une manière de lier les automates cellulaires à des objets de nature algébrique, ce qui permet de transférer des résultats d'un domaine vers l'autre. De même que la reconnaissance de langages par automates finis ou le calcul de fonctions par machines de Turing possèdent des caractérisations en terme de théorie logique, est-il possible de caractériser de façon logique les classes d'équivalence du groupage et, éventuellement, d'établir un lien entre la taille des formules obtenues et la position des classes d'équivalence dans le groupage ?
- En étudiant les problèmes de décision liés à des familles d'automates cellulaires particulières, nous avons été confrontés au problème plus général suivant. Étant données deux propriétés P_1 et P_2 des automates cellulaires, il est parfois possible d'associer récursivement à tout automate cellulaire vérifiant P_1 un automate cellulaire vérifiant P_1 et P_2 qui le simule et à tout automate cellulaire ne vérifiant pas P_1 un automate cellulaire ne vérifiant pas P_1 mais vérifiant P_2 qui le simule. Or, pour établir des réductions entre problèmes de décision, une possibilité est d'associer récursivement à tout automate cellulaire un automate cellulaire vérifiant P_2 et vérifiant P_1 si et seulement si l'automate cellulaire d'origine le vérifie. Existe-t-il une méthode générale pour obtenir ce dernier type de transformation à partir des deux décrites plus haut, ou existe-t-il des contre-exemples ?
- Bien que ce résultat ne figure pas dans cette thèse, nous avons démontré qu'il existe des automates cellulaires Turing universels simulant des machines de Turing à k têtes qui ne sont pas intrinsèquement universels et tels qu'un automate cellulaire de cette famille simule strictement les automates cellulaires de la famille possédant strictement moins de têtes. En conséquence, il existe une infinité de classes d'équivalence contenant des automates cellulaires Turing universels — toute classe d'équivalence contenant des automates cellulaires Turing universels ne contient que ce type d'automate cellulaire. Que dire des automates cellulaires Turing universels et de leur position dans le groupage en général ?
- Concernant l'universalité des automates cellulaires de dimension 1 à voisinage de von Neumann et 2 états, S. Wolfram [91] présente une vague et informelle descrip-

tion d'arguments en faveur d'une forme faible d'universalité Turing de l'automate cellulaire connu sous le nom de règle 110. Les arguments développés par S. Wolfram sont-ils extensibles en une démonstration formelle de l'universalité, éventuellement intrinsèque, de cet automate cellulaire ?

ANNEXE

A GROUPAGE CARRÉ

Nous présentons ici en détail le groupage et les résultats associés présentés par I. Rapaport [12] dans son mémoire de thèse. Afin de faciliter une lecture croisée avec le document original, les propositions sont numérotées à l'identique et nous suivons le même plan. Les démonstrations sont omises car d'une part elles ont déjà été présentées en détail dans le mémoire original et d'autre part une grande partie de ces résultats sont démontrés à l'aide de nouveaux outils dans le chapitre 2.

Sommaire

A.1 Définitions **109**
A.2 Propriétés élémentaires **110**
A.3 Le bas de l'ordre **111**
A.3.1 Automates cellulaires dynamiquement simples 111
A.3.2 Automates cellulaires algébriquement simples 113
A.4 Chaînes infinies bornées **114**
A.5 Chaînes infinies non-bornées **115**

A.1 Définitions

On considère uniquement les automates cellulaires de dimension 1 à voisinage de von Neumann. Aussi, pour alléger les notations, on note (Q, δ) l'automate cellulaire $(\mathbb{Z}, Q, [\![-1, 1]\!], \delta)$ et G_δ sa règle globale de transition. Dans un diagramme espace-temps d'un tel automate cellulaire, le k-uplet des états d'un segment de k cellules à l'instant t dépend des états de $2t + k$ cellules à l'instant 0 comme représenté sur la figure A.1.

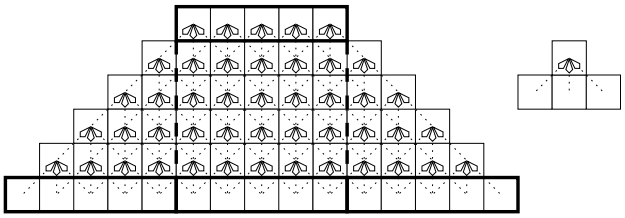


FIG. A.1 – Diagramme de dépendances d'un automate cellulaire

L'itéré n fois de la règle locale de transition est défini récursivement par $\delta^1 = \delta$ et, pour tout entier $n \in \mathbb{N}$,

$$\begin{aligned}\delta^{n+1}(x_{-n-1}, \ldots, x_{n+1}) = \delta(&\delta^n(x_{-n-1}, \ldots, x_{n-1}),\\ &\delta^n(x_{-n} \quad, \ldots, x_n \quad),\\ &\delta^n(x_{-n+1}, \ldots, x_{n+1}))\quad .\end{aligned}$$

En s'inspirant des dépendances dans les diagrammes espace-temps et des considérations géométriques présentées lors de la discussion sur les classifications et le groupage, on

définit la n-ième puissance $(Q,\delta)^n$ de l'automate cellulaire (Q,δ) par $(Q,\delta)^n = (Q^n, \delta^n_\square)$ où $\delta^n_\square$ est défini pour tout triplet de n-uplets d'états par

$$\begin{aligned}\delta^n_\square((x_1,\ldots,x_n),(x_{n+1},\ldots,x_{2n}),(x_{2n+1},\ldots,x_{3n})) = \big(&\delta^n(x_1,\ldots,x_{2n+1}),\\ &\delta^n(x_2,\ldots,x_{2n+2}),\\ &\vdots\\ &\delta^n(x_n,\ldots,x_{3n})\big) \quad .\end{aligned}$$

Pour tout entier $n > 0$, les diagrammes espace-temps de tout automate cellulaire (Q,δ) sont mis en bijection avec les diagrammes espace-temps de sa n-ième puissance par l'opération de groupage.

► En effet, à toute évolution Δ de (Q,δ) on peut associer bijectivement la suite de configurations Δ' de $(Q,\delta)^n$ définie pour tout temps t et toute position p par $\Delta'^{(t)}_p = \left(\Delta^{(nt)}_{np},\ldots,\Delta^{(nt)}_{(n+1)p-1}\right)$ qui est, par définition de $\delta^n_\square$, une évolution de $(Q,\delta)^n$.

Une fois cette transformation définie, on introduit la relation de comparaison $\leqslant_\square$ entre automates cellulaires. Afin de fixer les termes, on dira qu'un automate cellulaire (Q,δ) est simulé par un automate cellulaire (Q',δ') si $(Q,\delta) \leqslant_\square (Q',\delta')$.

Définition 9. *Un automate cellulaire (Q,δ) est simulé par un automate cellulaire (Q',δ'), noté $(Q,\delta) \leqslant_\square (Q',\delta')$, s'il existe deux puissances m et n telles que $(Q,\delta)^m \subseteq (Q',\delta')^n$.*

La première étape est de montrer qu'il s'agit d'un préordre. De la démonstration proposée par I. Rapaport pour montrer que $\leqslant_\square$ est un préordre, on peut retenir deux lemmes clefs.

Lemme 2. *Soient (Q,δ) et (Q',δ') deux automates cellulaires. Si $(Q,\delta) \subseteq (Q',\delta')$ alors, pour toute puissance n, on a $(Q,\delta)^n \subseteq (Q',\delta')^n$.*

Lemme 3. *Soit (Q,δ) un automate cellulaire et m et n deux puissances. Alors, on a $((Q,\delta)^m)^n \cong (Q,\delta)^{mn}$.*

Ces deux lemmes apportent des informations structurelles sur les liens entre les automates cellulaires, leurs puissances et la relation sous-automate. La démonstration de la proposition suivante utilise ces propriétés :

- la relation sous-automate est une relation d'ordre partiel ;
- la relation sous-automate est préservée par passage à une même puissance ;
- l'opérateur de passage à la puissance est compatible avec la structure de monoïde commutatif de $(\mathbb{N}^*,\cdot)$.

Proposition 1. *La relation $\leqslant_\square$ est une relation de préordre.*

Dans la suite, on note $\sim_\square$ la relation d'équivalence associée à $\leqslant_\square$ et $<_\square$ la relation de comparaison stricte. On s'intéresse alors à la classification induite par les classes d'équivalences ordonnées par l'ordre induit.

A.2 Propriétés élémentaires

Proposition 2. *Toutes les puissances d'un automate cellulaire (Q,δ) sont équivalentes (*i.e. *pour toutes puissances i et j, on a $(Q,\delta)^i \sim_\square (Q,\delta)^j$).*

Après avoir vérifié que toutes les puissances d'un automate cellulaire sont dans la même classe d'équivalence, on établit quelques propriétés des minima et maxima de la classification. Notons $\perp$ l'unique automate cellulaire à un seul état. C'est l'automate cellulaire dont les évolutions sont les plus simples, c'est un minimum global.

Proposition 3. *L'automate cellulaire $\perp$ est seul dans sa classe d'équivalence, qui est le minimum global pour la relation $\leqslant_\square$.*

Le produit cartésien de deux automates cellulaires simule chacun de ces automates. L'idée est d'utiliser une configuration monochromatique périodique dans le temps sur une des composantes.

Proposition 4. *La relation $\leqslant_\square$ admet des maxima locaux. Pour toute famille finie d'automates cellulaires $\mathcal{A}_1,\ldots,\mathcal{A}_k$, l'automate cellulaire $\mathcal{B} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_k$ satisfait, pour tout indice $i \in [\![1, k]\!]$, la relation $\mathcal{A}_i \leqslant_\square \mathcal{B}$.*

A.3 Le bas de l'ordre

L'exploration continue par les classes d'équivalence situées juste au-dessus de $\perp$. On montre qu'il existe une infinité de tels automates cellulaires et que des automates cellulaires intuitivement simples en font partie.

A.3.1 Automates cellulaires dynamiquement simples

On considère tout d'abord des automates cellulaires dont les évolutions sont simples à décrire. Les plus simples d'entre eux sont certainement les automates cellulaires nilpotents, c'est-à-dire appartenant à l'ensemble $\mathrm{NIL} = \{(Q, \delta) \mid |Q| > 1 \wedge |\Omega(Q, \delta)| = 1\}$. Un représentant canonique de cette famille est l'automate cellulaire à n états

$$\mathbf{0}_n = ([\![0, n-1]\!], (x_l, x_m, x_r) \mapsto 0)$$

dont la règle globale de transition constante a toujours pour image la configuration 0-monochromatique. On montre que cette famille est une classe d'équivalence située au bas de l'ordre.

Proposition 5. *La classe d'équivalence de $\mathbf{0}_2$ est exactement l'ensemble des automates cellulaires nilpotents (*i.e. *pour tout automate cellulaire (Q, δ), on a*

$$(Q, \delta) \in \mathrm{NIL} \iff (Q, \delta) \sim_\square \mathbf{0}_2 \quad).$$

Proposition 6. *La classe d'équivalence* NIL *est au bas de l'ordre (*i.e. *pour tout automate cellulaire (Q, δ) différent de $\perp$, on a $(Q, \delta) \leqslant_\square \mathbf{0}_2 \Longrightarrow \mathbf{0}_2 \leqslant_\square (Q, \delta)$).*

► Pour démontrer les deux propositions précédentes, on utilise les deux faits suivants : d'une part toute puissance et tout sous-automate d'un automate cellulaire nilpotent est nilpotent, d'autre part si (Q, δ) est un automate cellulaire nilpotent de transitoire bornée par T alors $(Q, \delta)^T \cong \mathbf{0}_{Q^T}$ et pour tout n on a $\mathbf{0}_n \subseteq \mathbf{0}_{2^{\lceil \log_2 n \rceil}} \cong \mathbf{0}_2^{\lceil \log_2 n \rceil}$.

De ces propositions et de l'indécidabilité de la nilpotence de J. Kari [44], on déduit un résultat d'indécidabilité sur $\leqslant_\square$. Ce qui montre que cette classification, comme toutes les classifications intéressantes, est indécidable.

Corollaire 6. *Étant donné un automate cellulaire (Q,δ), savoir si $(Q,\delta) \leqslant_\square \mathbf{0}_2$ est indécidable.*

Trois autres familles d'automates cellulaires simples sont introduites : les automates cellulaires périodiques, les automates cellulaires périodiques composés avec une translation vers la gauche et les automates cellulaires périodiques composés avec une translation vers la droite.

$$\begin{aligned}
\mathrm{PER} &= \left\{(Q,\delta) \mid |Q| > 1 \wedge \forall C \in Q^{\mathbb{Z}}, \exists n \in \mathbb{N}^*, G_\delta^n(C) = C\right\} \\
L_{\mathrm{SHIFT}} &= \left\{(Q,\delta) \mid |Q| > 1 \wedge \forall C \in Q^{\mathbb{Z}}, \exists n \in \mathbb{N}^*, \forall i \in \mathbb{Z}, G_\delta^n(C)_i = C_{i+n}\right\} \\
R_{\mathrm{SHIFT}} &= \left\{(Q,\delta) \mid |Q| > 1 \wedge \forall C \in Q^{\mathbb{Z}}, \exists n \in \mathbb{N}^*, \forall i \in \mathbb{Z}, G_\delta^n(C)_i = C_{i-n}\right\}
\end{aligned}$$

Remarquons que ces trois ensembles sont définis par une formule du type

$$|Q| > 1 \wedge \forall C \in Q^{\mathbb{Z}}, \exists n \in \mathbb{N}^*, \Phi(C,n)$$

pour une sous-formule Φ donnée. En fait, par le biais des configurations univers, on peut montrer que, dans chacun de ces trois cas, la formule

$$|Q| > 1 \wedge \exists n \in \mathbb{N}^*, \forall C \in Q^{\mathbb{Z}}, \Phi(C,n)$$

définit à chaque fois le même ensemble. Pour chacune de ces familles, on introduit un représentant canonique :

$$\begin{aligned}
\boldsymbol{\iota}_n &= (\llbracket 0, n-1 \rrbracket, (x_l, x_m, x_r) \mapsto x_m) \\
\boldsymbol{\sigma}_n &= (\llbracket 0, n-1 \rrbracket, (x_l, x_m, x_r) \mapsto x_l) \\
\boldsymbol{\sigma}_n^{-1} &= (\llbracket 0, n-1 \rrbracket, (x_l, x_m, x_r) \mapsto x_r)
\end{aligned}$$

On montre alors, de manière similaire à ce qui a été fait pour les automates cellulaires nilpotents, que chacune de ces familles est en fait une classe d'équivalence et que celle-ci est située au bas de l'ordre.

Proposition 7. *Les classes d'équivalence des automates cellulaires $\boldsymbol{\iota}_2$, $\boldsymbol{\sigma}_2$ et $\boldsymbol{\sigma}_2^{-1}$ correspondent respectivement aux ensembles d'automates cellulaires* PER*, R_{SHIFT} et L_{SHIFT} (*i.e. *pour tout automate cellulaire (Q,δ) différent de $\perp$, on a*

$$\begin{aligned}
\boldsymbol{\iota}_2 \sim_\square (Q,\delta) &\iff (Q,\delta) \in \mathrm{PER} \\
\boldsymbol{\sigma}_2 \sim_\square (Q,\delta) &\iff (Q,\delta) \in R_{\mathrm{SHIFT}} \\
\boldsymbol{\sigma}_2^{-1} \sim_\square (Q,\delta) &\iff (Q,\delta) \in L_{\mathrm{SHIFT}} \quad).
\end{aligned}$$

Proposition 8. *Les classes d'équivalence* PER*, R_{SHIFT} et L_{SHIFT} appartiennent au bas de l'ordre (*i.e. *pour tout automate cellulaire (Q,δ) différent de $\perp$, on a*

$$\begin{aligned}
(Q,\delta) \leqslant_\square \boldsymbol{\iota}_2 &\implies \boldsymbol{\iota}_2 \leqslant_\square (Q,\delta) \\
(Q,\delta) \leqslant_\square \boldsymbol{\sigma}_2 &\implies \boldsymbol{\sigma}_2 \leqslant_\square (Q,\delta) \\
(Q,\delta) \leqslant_\square \boldsymbol{\sigma}_2^{-1} &\implies \boldsymbol{\sigma}_2^{-1} \leqslant_\square (Q,\delta) \quad).
\end{aligned}$$

Corollaire 7. *Les classes d'équivalence* NIL*,* PER*, R_{SHIFT} et L_{SHIFT} sont deux à deux incomparables.*

A.3.2 Automates cellulaires algébriquement simples

Pour montrer qu'il existe une infinité de classes d'équivalence au bas de l'ordre, on introduit une famille infinie d'automates cellulaires algébriquement simples. Pour tout entier $p > 1$, on considère le groupe additif abélien des entiers relatifs modulo p noté $(\mathbb{Z}_p, +)$ et, pour tout entier non-nul n, le groupe produit canonique $\left(\mathbb{Z}_p^n, +\right)$. On définit alors l'automate cellulaire $(\mathbb{Z}_p, \oplus)$ où, pour tout triplet (x, y, z) d'éléments de $\mathbb{Z}_p$, on pose $\oplus(x, y, z) = x + y + z$. De même, on définit $\left(\mathbb{Z}_p^n, \oplus\right)$. Pour tout entier $n > 0$, on note $(\mathbb{Z}_p, \oplus)^n = \left(\mathbb{Z}_p^n, \oplus_\square^n\right)$ la puissance n-ième de $(\mathbb{Z}_p, \oplus)$. On remarquera que $\left(\mathbb{Z}_p^n, \oplus\right)$ est différent de $\left(\mathbb{Z}_p^n, \oplus_\square^n\right)$. On commence par rappeler un résultat classique d'algèbre.

Proposition 9. *Soient p et n deux entiers non nuls tels que $p > 1$. Soit X un sous-ensemble non vide de $\mathbb{Z}_p^n$. Si X est stable par l'opérateur $+$, alors $(X, +)$ est un sous-groupe de $\left(\mathbb{Z}_p^n, +\right)$ dont l'ordre divise p^n. De plus, si p est premier X est similaire à un produit de la forme $\prod_{i=1}^n X_i$ où, pour tout i, X_i est égal à $\mathbb{Z}_p$ ou à $\{0\}$.*

Pour montrer que chaque $(\mathbb{Z}_p, \oplus)$ avec p premier est dans une classe d'équivalence différente, on montre que les puissances et leurs sous-automates ont de bonnes propriétés algébriques, en particulier ce sont des sous-groupes des puissances des groupes produits de l'ensemble d'états.

Proposition 10. *Soient $p, n \in \mathbb{N}^*$ avec $p > 1$. Pour tout $x_1, x_2, x_3, y_1, y_2, y_3 \in \mathbb{Z}_p^n$, on a*

$$\oplus_\square^n (x_1 + y_1, x_2 + y_2, x_3 + y_3) = \oplus_\square^n (x_1, x_2, x_3) + \oplus_\square^n (y_1, y_2, y_3) \quad .$$

Proposition 11. *Soit (Q, δ) un automate cellulaire et p un entier tel que $p > 1$. Si $(Q, \delta) \leqslant_\square (\mathbb{Z}_p, \oplus)$ alors il existe deux puissances $m, n \in \mathbb{N}^*$ et une application injective $\varphi : Q^m \to \mathbb{Z}_p^n$ vérifiant $\overline{\varphi} \circ \delta_\square^m = \oplus_\square^n \circ \overline{\varphi}$ et telle que $0 \in \varphi(Q^m)$.*

Proposition 12. *Soient $p, n \in \mathbb{N}^*$ avec $p > 1$. Soit X un sous-ensemble de $\mathbb{Z}_p^n$ contenant 0 et (Q, δ) un automate cellulaire. Si $(X, \oplus_\square^n) \subseteq (\mathbb{Z}_p^n, \oplus_\square^n)$ alors $(X, +)$ est un sous-groupe de $(\mathbb{Z}_p^n, +)$.*

On peut remarquer que si $a \mid b$ alors $(\mathbb{Z}_a, \oplus) \subseteq (\mathbb{Z}_b, \oplus)$ (il suffit de poser $\varphi(x) = \frac{b}{a}x$).

Corollaire 8. *Soient p et q deux entiers premiers. Si $p \neq q$ alors $(\mathbb{Z}_p, \oplus) \not\leqslant_\square (\mathbb{Z}_q, \oplus)$.*

Par des considérations sur les diagrammes espace-temps et quelques propositions très techniques, on montre que toutes ces classes d'équivalence sont en bas de l'ordre.

Proposition 13. *Soit n un entier non nul et p un nombre premier :*

- $\left(\mathbb{Z}_p^{p^n}, \oplus\right) \cong \left(\mathbb{Z}_p, \oplus\right)^{p^n}$;
- *Pour tout sous-ensemble X de $\mathbb{Z}_p^{p^n}$ à au moins deux éléments contenant 0, on a*

$$\left(X, \oplus_\square^{p^n}\right) \subseteq \left(\mathbb{Z}_p^{p^n}, \oplus_\square^{p^{p^n}}\right) \Longrightarrow \left(\mathbb{Z}_p, \oplus\right) \subseteq \left(X, \oplus_\square^{p^n}\right) \quad .$$

Proposition 14. *Soit n un entier non nul et p un nombre premier. Soit X un sous-ensemble de $\mathbb{Z}_p^n$ à au moins deux éléments contenant 0. Alors, on a*

$$\left(X, \oplus_\square^n\right) \subseteq \left(\mathbb{Z}_p^n, \oplus_\square^n\right) \Longrightarrow \exists m \in \mathbb{N}^*, \left(\mathbb{Z}_p^m, \oplus_\square^m\right) \subseteq \left(X, \oplus_\square^n\right) \quad .$$

Corollaire 9. *Pour tout nombre premier p, la classe d'équivalence de l'automate cellulaire $(\mathbb{Z}_p, \oplus)$ est au bas de l'ordre (*i.e. *pour tout automate cellulaire (Q, δ) différent de $\perp$, on a $(Q, \delta) \leqslant_\square (\mathbb{Z}_p, \oplus) \Longrightarrow (\mathbb{Z}_p, \oplus) \leqslant_\square (Q, \delta)$).*

Ainsi s'achève l'étude du bas de l'ordre. En particulier, on a montré que la classe d'équivalence du minimum global $\perp$ possède un degré sortant infini. La figure A.2 synthétise les connaissances acquises concernant le bas de l'ordre.

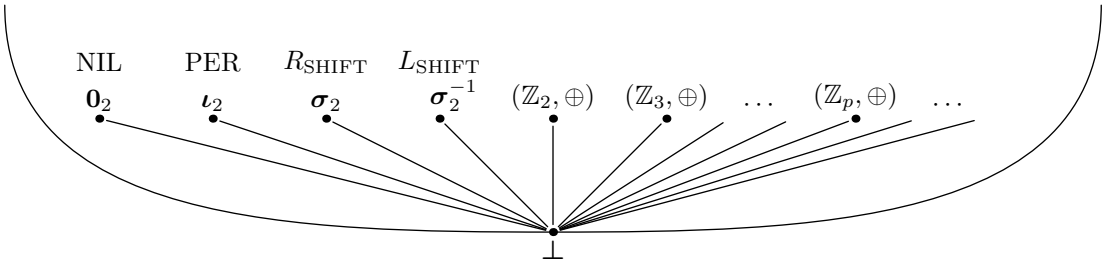


FIG. A.2 – Le bas de l'ordre $\leqslant_\square$: classes et représentants

A.4 Chaînes infinies bornées

L'étape suivante dans l'étude de la classification consiste en l'exhibition de deux chaînes croissantes infinies bornées de classes d'équivalences. Pour cela, on introduit les familles d'automates cellulaires suivantes indexées par un entier $n > 1$.

$$\begin{aligned}
\boldsymbol{\eta}_n &= \left([\![0, n-1]\!], (x_l, x_m, x_r) \mapsto \begin{cases} x_l & \text{si } x_l = x_m = x_r \\ 0 & \text{sinon} \end{cases} \right) \\
\boldsymbol{\mu}_n &= ([\![0, n-1]\!], (x_l, x_m, x_r) \mapsto \min\{x_l, x_m, x_r\})
\end{aligned}$$

Proposition 15. *Les automates cellulaires nilpotents sont strictement en dessous de $\boldsymbol{\eta}_2$ et $\boldsymbol{\mu}_2$ (*i.e. $\mathbf{0}_2 <_\square \boldsymbol{\eta}_2$ *et* $\mathbf{0}_2 <_\square \boldsymbol{\mu}_2$*).*

On montre alors que les classes d'équivalences de ces automates cellulaires forment deux chaînes strictement croissantes et incomparables entre elles.

Proposition 16. *Pour tout entier $n > 1$, on a $\boldsymbol{\eta}_n <_\square \boldsymbol{\eta}_{n+1}$ et $\boldsymbol{\mu}_n <_\square \boldsymbol{\mu}_{n+1}$.*

Proposition 17. *Pour tout couple d'entiers $m > 2$ et $n > 2$, on a $\boldsymbol{\mu}_m \nleqslant_\square \boldsymbol{\eta}_n$.*

Proposition 18. *Pour tout couple d'entiers $m > 2$ et $n > 2$, on a $\boldsymbol{\eta}_m \nleqslant_\square \boldsymbol{\mu}_n$.*

Afin de borner ces chaînes, I. Rapaport introduit un automate cellulaire spécifique que nous noterons $\boldsymbol{\vartheta}$. Nous omettons les détails de la construction qui sont inutiles pour notre discours. Cet automate cellulaire majore les deux chaînes.

Proposition 19. *Pour tout entier $n > 1$, on a $\boldsymbol{\eta}_n <_\square \boldsymbol{\vartheta}$.*

Proposition 20. *Pour tout entier $n > 1$, on a $\boldsymbol{\mu}_n <_\square \boldsymbol{\vartheta}$.*

La figure A.3 synthétise les informations concernant ces deux chaînes infinies bornées.

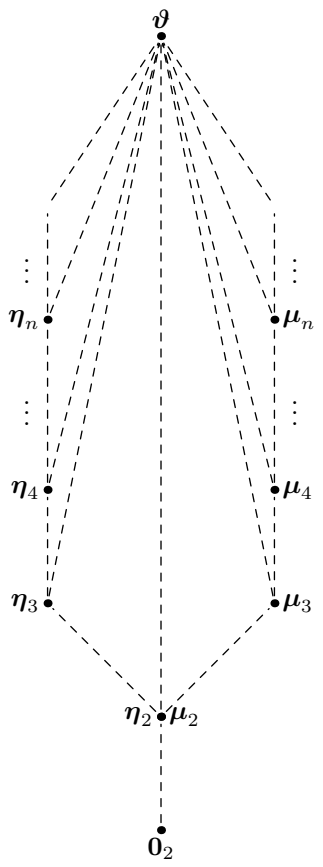


FIG. A.3 – Chaînes infinies bornées pour l'ordre $\leqslant_\square$

A.5 Chaînes infinies non-bornées

Alors que les chaînes infinies précédentes sont bornées, il est possible d'en exhiber de non bornées et d'en déduire que la classification n'admet pas de maximum. On introduit la famille suivante d'automates cellulaires, indexée par un entier $n > 1$:

$$\boldsymbol{\Delta}_n = \left([\![0, n-1]\!], (x_l, x_m, x_r) \mapsto \begin{cases} x_l & \text{si } x_l = x_r \\ x_m & \text{sinon} \end{cases} \right) .$$

On commence par situer ces automates cellulaires par rapport aux précédents et on montre une propriété étonnante : $\boldsymbol{\Delta}_n$ n'est jamais sous-automate d'une puissance d'un automate cellulaire à moins de n états.

Proposition 21. $\mathbf{0}_2 <_\square \boldsymbol{\mu}_2 <_\square \boldsymbol{\Delta}_2$.

Proposition 22. *Soit (Q, δ) un automate cellulaire et $n \in \mathbb{N}^*$ un entier. Si $|Q| < n$ alors, pour tout $i \in \mathbb{N}^*$ on a $\boldsymbol{\Delta}_n \nsubseteq (Q, \delta)^i$.*

Afin d'utiliser cette proposition, on montre comment trouver pour tout automate cellulaire un second automate cellulaire dont le premier est sous-automate de toutes les puissances.

Lemme 4. *Pour tout automate cellulaire (Q, δ), il existe une version normalisée $(Q, \delta)^*$ de cet automate cellulaire telle que pour tout entier $i \in \mathbb{N}^*$, on a $(Q, \delta) \subseteq ((Q, \delta)^*)^i$.*

Pour la version normalisée $(Q,\delta)^*$ de l'automate cellulaire (Q,δ), on peut choisir l'automate cellulaire partitionné $\left(\mathbb{Z}, (Q \cup \{\bullet\})^3, [\![-1,1]\!], \psi\right)$ où l'application ψ est définie, pour tout triplet d'états (x_l, x_m, x_r), par

$$\psi(x_l, x_m, x_r) = \begin{cases} (x_l, x_m, x_r) & \text{si } \bullet \in \{x_l, x_m, x_r\}\,; \\ (x, x, x) & \text{où } x = \delta(x_l, x_m, x_r) \text{ sinon.} \end{cases}$$

Pour montrer que (Q,δ) est un sous-automate de $((Q,\delta)^*)^i$, il suffit alors de prendre pour application injective $\varphi : x \mapsto ((x,x,x), \underbrace{(\bullet,\bullet,\bullet), \ldots, (\bullet,\bullet,\bullet)}_{i-1})$.

Proposition 23. *Soit (Q,δ) un automate cellulaire et $n \in \mathbb{N}^*$ un entier. Si $|Q| < n$ alors on a $\mathbf{\Delta}_n^* \not\leqslant_\square (Q,\delta)$.*

Corollaire 10. *L'ordre $\leqslant_\square$ n'admet pas de maximum.*

Grâce à ce corollaire, on peut montrer que les $\mathbf{\Delta}_n$ forment des chaînes infinies. En combinant tous ces résultats avec le produit cartésien, on montre qu'il n'existe aucun élément maximal pour l'ordre.

Proposition 24. *Pour tout entier $n \in \mathbb{N}^*$, on a $\mathbf{\Delta}_n^* <_\square \mathbf{\Delta}_{(n+1)^3+1}^*$.*

Proposition 25. *Pour tout automate cellulaire (Q,δ) on a $(Q,\delta) <_\square (Q,\delta) \times \mathbf{\Delta}_{|Q|+1}^*$.*

Corollaire 11. *L'ordre $\leqslant_\square$ n'admet pas d'élément maximal.*

La figure A.4 synthétise les informations concernant la chaîne infinie extraite de la famille d'automates cellulaires $(\mathbf{\Delta}_n^*)_{n\in\mathbb{N}}$.

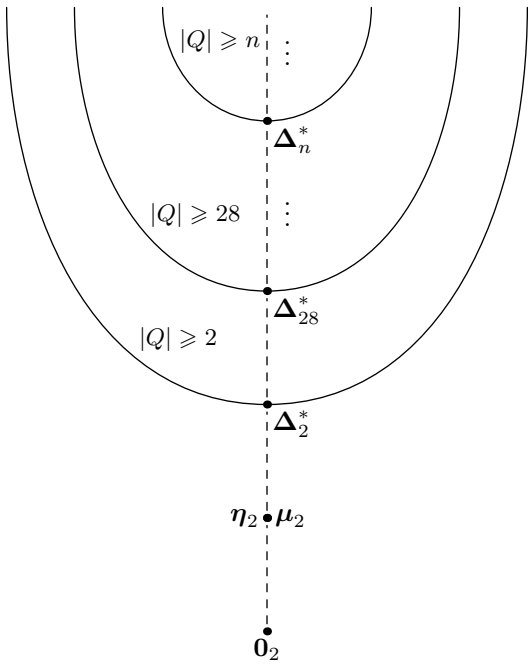


FIG. A.4 – Une chaîne infinie non-bornée pour l'ordre $\leqslant_\square$

BIBLIOGRAPHIE

Bibliographie personnelle

[1] N. Ollinger, Two-states bilinear intrinsically universal cellular automata, *Fundamentals of computation theory (Riga, Latvia, 2001)*, edité par R. Freivalds, tome 2138 de *Lecture Notes in Computer Science* (Springer, Berlin, 2001).

[2] N. Ollinger, The quest for small universal cellular automata, *International Colloquium on Automata, languages and programming (Málaga, Spain, 2002)*, edité par P. Widmayer, F. Triguero, R. Morales, M. Hennessy, S. Eidenbenz et R. Conejo, tome 2380 de *Lecture Notes in Computer Science* (Springer, Berlin, 2002).

[3] N. Ollinger, The intrinsic universality problem of one-dimensional cellular automata, *Symposium on Theoretical Aspects of Computer Science (Berlin, Germany, 2003)*, Lecture Notes in Computer Science (Springer, Berlin, 2003), (to appear).

Thèses de doctorat

[4] E. R. Banks, *Information processing and transmission in cellular automata*, Thèse de doctorat, Massachusetts Institute of Technology, 1971.

[5] J. Kari, *Decision Problems Concerning Cellular Automata*, Thèse de doctorat, University of Turku, 1989.

[6] B. Martin, *Construction modulaire d'automates cellulaires*, Thèse de doctorat, École Normale Supérieure de Lyon, 1993.

[7] B. Durand, *Cellular automata: reversibility and complexity*, Thèse de doctorat, École Normale Supérieure de Lyon, 1994.

[8] Z. Róka, *Cellular automata on Cayley graphs*, Thèse de doctorat, École Normale Supérieure de Lyon, 1994.

[9] J. Durand-Lose, *Automates Cellulaires, Automates à Partitions et Tas de Sable*, Thèse de doctorat, Université Bordeaux I, 1996.

[10] O. Heen, *Économie de ressources sur Automates Cellulaires*, Thèse de doctorat, Université Paris 7, 1996.

[11] E. Formenti, *Cellular automata and chaos: from topology to Kolmogorov complexity*, Thèse de doctorat, École Normale Supérieure de Lyon, 1998.

[12] I. Rapaport, *Inducing an order on cellular automata by a grouping operation*, Thèse de doctorat, École Normale Supérieure de Lyon, 1998.

[13] B. Martin, II, *Automates cellulaires, information et chaos*, Thèse de doctorat, École Normale Supérieure de Lyon, 2001.

[14] J. Cervelle, *Complexité structurelle et algorithmique des pavages et des automates cellulaires*, Thèse de doctorat, Université de Provence, 2002.

Articles de journaux et actes de conférences

[15] C. E. Shannon, A Universal Turing Machine with Two Internal States, *Automata Studies*, edité par C. E. Shannon et J. McCarthy (Princeton University Press, Princeton, 1956), (pp. 157–165).

[16] E. F. Moore, Machine models of self-reproduction, *Proceedings of Symposia in Applied Mathematics*, tome 14 (American Mathematical Society, 1962), (pp. 17–33).

[17] J. Myhill, The converse of Moore's Garden-of-Eden theorem, *Proceedings of the American Mathematical Society*, tome 14 (American Mathematical Society, 1963), (pp. 658–686).

[18] P. C. Fischer, Generation of Primes by a One-Dimensional Real-Time Iterative Array, *Journal of the ACM*, **12**, no. 3 (1965) 388–394.

[19] G. A. Hedlund, Endormorphisms and automorphisms of the shift dynamical system, *Mathematical Systems Theory*, **3** (1969) 320–375.

[20] E. R. Banks, Universality in cellular automata, *Symposium on Switching and Automata Theory (Santa Monica, California, 1970)* (IEEE, 1970), (pp. 194–215).

[21] R. M. Robinson, Undecidability and nonperiodicity for tilings of the plane, *Inventiones Mathematicae*, **12** (1971) 177–209.

[22] A. R. Smith, III, Simple computation-universal cellular spaces, *Journal of the ACM*, **18** (1971) 339–353.

[23] S. Amoroso et Y. N. Patt, Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures, *Journal of Computer and System Sciences*, **6** (1972) 448–464.

[24] D. Richardson, Tessellations with local transformations, *Journal of Computer and System Sciences*, **6** (1972) 373–388.

[25] A. R. Smith, III, Real-time language recognition by one-dimensional cellular automata, *Journal of Computer and System Sciences*, **6** (1972) 233–253.

[26] C. Choffrut et K. Čulik, II, On real-time cellular automata and trellis automata, *Acta Informatica*, **21**, no. 4 (1984) 393–407.

[27] C. Langton, Self-Reproduction in Cellular Automata, *Physica D*, **10**, no. 1-2 (1984) 135–144.

[28] O. Martin, A. M. Odlyzko et S. Wolfram, Algebraic properties of cellular automata, *Communications in Mathematical Physics*, **93**, no. 2 (1984) 219–258.

[29] S. Wolfram, Computation theory of cellular automata, *Communications in Mathematical Physics*, **96**, no. 1 (1984) 15–57.

[30] S. Wolfram, Universality and complexity in cellular automata, *Physica D. Nonlinear Phenomena*, **10**, no. 1-2 (1984) 1–35, cellular automata (Los Alamos, N.M., 1983).

[31] O. H. Ibarra, S. M. Kim et S. Moran, Sequential machine characterizations of trellis and cellular automata and applications, *SIAM Journal on Computing*, **14**, no. 2 (1985) 426–447.

[32] O. H. Ibarra, M. A. Palis et S. M. Kim, Fast parallel language recognition by cellular automata, *Theoretical Computer Science*, **41**, no. 2-3 (1985) 231–246 (1986).

[33] S. Wolfram, Twenty problems in the theory of cellular automata, *Physica Scripta*, **T9** (1985) 170–183, physics of chaos and related problems (Gräftåvallen, 1984).

[34] J. Albert et K. Čulik, II, A simple universal cellular automaton and its one-way and totalistic version, *Complex Systems*, **1**, no. 1 (1987) 1–16.

[35] R. H. Gilman, Classes of linear automata, *Ergodic Theory and Dynamical Systems*, **7**, no. 1 (1987) 105–118.

[36] K. Čulik, II, On invertible cellular automata, *Complex Systems*, **1**, no. 6 (1987) 1035–1044.

[37] K. Čulik, II et S. Yu, Undecidability of CA classification schemes, *Complex Systems*, **2**, no. 2 (1988) 177–190.

[38] K. Čulik, II, J. Pachl et S. Yu, On the limit sets of cellular automata, *SIAM Journal on Computing*, **18**, no. 4 (1989) 831–842.

[39] J. Kari, Reversibility of 2D cellular automata is undecidable, *Physica D. Nonlinear Phenomena*, **45**, no. 1-3 (1990) 379–385, cellular automata: theory and experiment (Los Alamos, NM, 1989).

[40] K. Lindgren et M. G. Nordahl, Universal computation in simple one-dimensional cellular automata, *Complex Systems*, **4**, no. 3 (1990) 299–318.

[41] K. Čulik, II, L. P. Hurd et S. Yu, Computation theoretic aspects of cellular automata, *Physica D. Nonlinear Phenomena*, **45**, no. 1-3 (1990) 357–378, cellular automata: theory and experiment (Los Alamos, NM, 1989).

[42] K. Sutner, Classifying circular cellular automata, *Physica D. Nonlinear Phenomena*, **45**, no. 1-3 (1990) 386–395, cellular automata : theory and experiment (Los Alamos, NM, 1989).

[43] K. Sutner, de Bruijn graphs and linear cellular automata, *Complex Systems*, **5**, no. 1 (1991) 19–30.

[44] J. Kari, The nilpotency problem of one-dimensional cellular automata, *SIAM Journal on Computing*, **21**, no. 3 (1992) 571–586.

[45] J. Mazoyer et N. Reimen, A linear speed-up theorem for cellular automata, *Theoretical Computer Science*, **101**, no. 1 (1992) 59–98, logic and applications to computer science (Paris, 1990/1991).

[46] R. Bartlett et M. Garzon, Monomial cellular automata, *Complex Systems*, **7**, no. 5 (1993) 367–388.

[47] B. Durand, Global properties of 2D cellular automata: some complexity results, *Mathematical foundations of computer science (Gdańsk, 1993)* (Springer, Berlin, 1993), (pp. 433–441).

[48] B. Durand, Undecidability of the surjectivity problem for 2D cellular automata : a simplified proof, *Fundamentals of computation theory (Szeged, 1993)* (Springer, Berlin, 1993), (pp. 204–211).

[49] B. Durand, Inversion of 2D cellular automata: some complexity results, *Theoretical Computer Science*, **134**, no. 2 (1994) 387–401.

[50] B. Durand, The surjectivity problem for 2D cellular automata, *Journal of Computer and System Sciences*, **49**, no. 3 (1994) 718–725.

[51] B. Durand et J. Mazoyer, Growing patterns in one-dimensional cellular automata, *Complex Systems*, **8**, no. 6 (1994) 419–434.

[52] J. Kari, Reversibility and surjectivity problems of cellular automata, *Journal of Computer and System Sciences*, **48**, no. 1 (1994) 149–182.

[53] J. Kari, Rice's theorem for the limit sets of cellular automata, *Theoretical Computer Science*, **127**, no. 2 (1994) 229–254.

[54] B. Martin, A universal cellular automaton in quasilinear time and its *S-m-n* form, *Theoretical Computer Science*, **123**, no. 2 (1994) 199–237.

[55] Z. Róka, One-way cellular automata on Cayley graphs, *Theoretical Computer Science*, **132**, no. 1-2 (1994) 259–290.

[56] R. Bartlett et M. Garzon, Bilinear cellular automata, *Complex Systems*, **9**, no. 6 (1995) 455–476.

[57] K. Morita, Reversible simulation of one-dimensional irreversible cellular automata, *Theoretical Computer Science*, **148**, no. 1 (1995) 157–163.

[58] J. Kari, Representation of reversible cellular automata with block permutations, *Mathematical Systems Theory*, **29**, no. 1 (1996) 47–61.

[59] J. Mazoyer, Computations on one-dimensional cellular automata, *Annals of Mathematics and Artificial Intelligence*, **16**, no. 1-4 (1996) 285–309.

[60] J. Durand-Lose, Intrinsic universality of a 1-dimensional reversible cellular automaton, *STACS 97 (Lübeck)*, tome 1200 de *Lecture Notes in Comput. Sci.* (Springer, Berlin, 1997), (pp. 439–450).

[61] P. Kůrka, Languages, equicontinuity and attractors in cellular automata, *Ergodic Theory and Dynamical Systems*, **17**, no. 2 (1997) 417–433.

[62] J. Mazoyer et I. Rapaport, Additive cellular automata over Z_p and the bottom of $(\mathrm{CA}, \leq)$, *Mathematical foundations of computer science (Brno, 1998)* (Springer, Berlin, 1998), (pp. 834–843).

[63] G. Cattaneo, E. Formenti et L. Margara, Topological definitions of deterministic chaos: applications to cellular automata dynamics, *Cellular automata (Saissac, 1996)* (Kluwer Acad. Publ., Dordrecht, 1999), (pp. 213–259).

[64] M. Delorme, An introduction to cellular automata: some basic definitions and concepts, *Cellular automata (Saissac, 1996)* (Kluwer Acad. Publ., Dordrecht, 1999), (pp. 5–49).

[65] M. Delorme et J. Mazoyer, Cellular automata as languages recognizers, *Cellular automata (Saissac, 1996)* (Kluwer Acad. Publ., Dordrecht, 1999), (pp. 153–179).

[66] B. Durand et Z. Róka, The game of life: universality revisited, *Cellular automata (Saissac, 1996)* (Kluwer Acad. Publ., Dordrecht, 1999), (pp. 51–74).

[67] O. H. Ibarra, Computational complexity of cellular automata: an overview, *Cellular automata (Saissac, 1996)* (Kluwer Acad. Publ., Dordrecht, 1999), (pp. 181–197).

[68] J. Mazoyer, Computations on cellular automata: some examples, *Cellular automata (Saissac, 1996)* (Kluwer Acad. Publ., Dordrecht, 1999), (pp. 77–118).

[69] J. Mazoyer, Computations on grids, *Cellular automata (Saissac, 1996)* (Kluwer Acad. Publ., Dordrecht, 1999), (pp. 119–149).

[70] J. Mazoyer et I. Rapaport, Global fixed point attractors of circular cellular automata and periodic tilings of the plane: undecidability results, *Discrete Mathematics*, **199**, no. 1-3 (1999) 103–122.

[71] J. Mazoyer et I. Rapaport, Inducing an order on cellular automata by a grouping operation, *Discrete Applied Mathematics*, **91**, no. 1-3 (1999) 177–196.

[72] J. Mazoyer et V. Terrier, Signals in one-dimensional cellular automata, *Theoretical Computer Science*, **217**, no. 1 (1999) 53–80, cellular automata (Milan, 1996).

[73] E. Rémila, An introduction to automata on graphs, *Cellular automata (Saissac, 1996)* (Kluwer Acad. Publ., Dordrecht, 1999), (pp. 345–363).

[74] Z. Róka, Simulations between cellular automata on Cayley graphs, *Theoretical Computer Science*, **225**, no. 1-2 (1999) 81–111.

[75] K. Imai et K. Morita, A computation-universal two-dimensional 8-state triangular reversible cellular automaton, *Theoretical Computer Science*, **231**, no. 2 (2000) 181–191, universal machines and computations (Metz, 1998).

[76] B. Martin, II, Apparent entropy of cellular automata, *Complex Systems*, **12**, no. 2 (2000) 135–155.

[77] J.-C. Dubacq, B. Durand et E. Formenti, Kolmogorov complexity and cellular automata classification, *Theoretical Computer Science*, **259**, no. 1-2 (2001) 271–285.

[78] M. Kudlek et Y. Rogozhin, New Small Universal Circular Post Machines, *Fundamentals of computation theory (Riga, 2001)*, edité par R. Freivalds (Springer, Berlin, 2001), (pp. 217–226).

[79] K. Morita et K. Imai, Number-conserving reversible cellular automata and their computation-universality, *Theoretical Informatics and Applications*, **35**, no. 3 (2001) 239–258.

Monographies

[80] J. von Neumann, *Theory of Self-Reproducing Automata* (University of Illinois Press, Urbana, Ill., 1966).

[81] M. Minsky, *Computation : Finite and Infinite Machines* (Prentice Hall, Englewoods Cliffs, 1967).

[82] E. F. Codd, *Cellular Automata* (Academic Press, New York, 1968).

[83] K. Zuse, *Rechnender Raum*, tome 1 de *Scriften zur Datenverarbeitung* (Friedrich Vieweg & Sohn, Braunschweig, 1969).

[84] A. W. Burks (éditeur), *Essays on cellular automata* (University of Illinois Press, Urbana, Ill., 1970).

[85] E. R. Berlekamp, J. H. Conway et R. K. Guy, *Winning ways for your mathematical plays. Vol. 2* (Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London, 1982), games in particular.

[86] S. Wolfram (éditeur), *Theory and applications of cellular automata* (World Scientific Publishing Co., Singapore, 1986), including selected papers 1983–1986.

[87] I. Wegener, *The Complexity of Boolean Functions* (B. G. Teubner, and John Wiley & Sons, 1987).

[88] M. Garzon, *Models of massive parallelism* (Springer-Verlag, Berlin, 1995), analysis of cellular automata and neural networks.

[89] J. H. Conway et N. J. A. Sloane, *Sphere packings, lattices and groups*, tome 290 de *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]* (Springer-Verlag, New York, 1999), troisième édition.

[90] M. Delorme et J. Mazoyer (éditeurs), *Cellular automata* (Kluwer Academic Publishers, Dordrecht, 1999), a parallel model, Papers from the Spring School of Theoretical Computer Science held in Saissac, 1996.

[91] S. Wolfram, *A New Kind of Science* (Wolfram Media, Inc., Champaign, Ill., 2002).

INDEX DES AUTEURS

Albert, J. 12, 50, 88, 90, **119**
Amoroso, S. 27, **118**

Banks, E. R. 12, 50, 88, 90, **117**, **118**
Bartlett, R. 94, **119**, **120**
Berlekamp, E. R. 90, **121**

Cattaneo, G. 36, **120**
Cervelle, J. 29, 36, **117**
Choffrut, C. 49, **118**
Codd, E. F. 90, **121**
Conway, J. H. 90, 107, **121**, **122**
Čulik, II, K. 12, 28, 29, 35, 49, 50, 88, 90, **118**, **119**

Delorme, M. **120**
Dubacq, J.-C. **121**
Durand, B. 26, 27, 50, 88, **117**, **119**, **120**, **121**
Durand-Lose, J. 88, **117**, **120**

Fischer, P. C. 11, 48, **118**
Formenti, E. 36, **117**, **120**, **121**

Garzon, M. 94, **119**, **120**, **122**
Gilman, R. H. 36, **119**
Guy, R. K. 90, **121**

Hedlund, G. A. 12, 25, 26, 32, **118**
Heen, O. **117**
Hurd, L. P. **119**

Ibarra, O. H. 49, **118**, **120**
Imai, K. 88, **121**

Kari, J. 12, 27, 29, 31, 111, **117**, **119**, **120**
Kim, S. M. 49, **118**
Kudlek, M. 90, **121**
Kůrka, P. 36, **120**

Langton, C. **118**
Lindgren, K. 90, **119**

Margara, L. 36, **120**
Martin, B. 12, 50, 88, **117**, **120**
Martin, II, B. 12, 36, **117**, **121**
Martin, O. **118**
Mazoyer, J. 29, 91, **119**, **120**, **121**
Minsky, M. **121**
Moore, E. F. 11, 26, **118**
Moran, S. 49, **118**
Morita, K. 88, **120**, **121**
Myhill, J. 11, 26, **118**

Neumann (von), J. 11, 19, 87, 88, 90, **121**
Nordahl, M. G. 90, **119**

Odlyzko, A. M. **118**
Ollinger, N. **117**

Pachl, J. 12, 28, 29, **119**
Palis, M. A. **118**
Patt, Y. N. 27, **118**

Rapaport, I. 12, 29, 39, 45, 88, 91, 109, **117**, **120**, **121**
Reimen, N. **119**
Rémila, E. **121**
Richardson, D. 25, **118**
Robinson, R. M. 27, **118**
Rogozhin, Y. 90, **121**
Róka, Z. 50, 88, **117**, **120**, **121**

Shannon, C. E. 90, **118**
Sloane, N. J. A. 107, **122**
Smith, III, A. R. 90, **118**
Sutner, K. 27, 29, **119**

Terrier, V. **121**

Wegener, I. 98, **121**
Wolfram, S. 12, 33, 34, 90, 107, **118**, **119**, **122**

Yu, S. 12, 28, 29, 35, **119**

Zuse, K. 11, 19, **121**

INDEX

arrêt, 45, 88
automate cellulaire, 20
  6 états, 97
  autarcique, 30
  bijectif, 27
  bilinéaire, 24, 94
  chaudière, 92
  conservatif, 81
  injectif, 26, 80
  linéaire, 24
  nilpotent, 28, 29, 49, 76, 80
    périodique, 29
  périodique, 80
  partiel, 48
  partitionné, 23
  polynomial, 24, 81
  puissance, voir puissance
  réversible, 26, 31, 86, 88
  surjectif, 26, 80, 86
  totalistique, 80
  transformé, 50
  unidirectionnel, 48, 80
  universel, voir universalité
axiome
  combinaison, 52
  compatibilité, 52
  divisibilité, 52
  proximité, 53
  surjectivité, 53
  transitivité, 53

borne supérieure, 85

caractérisation
  par clôture, 30
  topologique, 25
cellule, 20
chaîne infinie, 75
classification, 33
  algébrique, 36
  calculabilité, 45
  expérimentale, 33
  topologique, 36
codage, 87
configuration, 20, 24
  finie, 24
  périodique, 24
  ultimement périodique, 25
  univers, 25

décodage, 87
découpage, voir transformation
degré Turing, 45
demi-treillis, 82, 85
  idéal, 85
diagramme espace-temps, 20
  transformation, 57, 83
dimension, 20

ensemble d'états, voir état
ensemble limite, 28
espace des phases, 72
état, 20
  ensemble, 20, 22
  quiescent, 24
  renommage, voir isomorphisme
évolution, 20, 24

fonction booléenne, 98

graphe, voir espace des phases
groupage
  abstrait, 51
  axiomatisation, 67, 70, 85
  axiome, voir axiome
  carré, 45, 109
  comparaison, 73, 85
  extension, 71, 85
  idéal, 85
    non-principal, 86
    principal, 86

jardin d'Eden, 26

machine de Turing, 29, 89
macro-cellule, 89
maximum
  global, 77

local, 78
minimum global, 76

objet abstrait, 51
opérateurs géométriques, 59
opération
composition, 30, 52
produit cartésien, 30
transformation, 52
ordre
bas, 76
haut, 77

pavabilité, 27, 29
pavage, 60
plongement, 60
monoïde, 61
restriction, 68
point fixe, 29
problème de décision
CA-1D-NIL-PER, 91
CA-1D-UNIV, 91
CA-LEFT-POW, 79
CA-RIGHT-POW, 79
CA-UNIV, 79
propriété
élémentaire, 53
asymptotique, 27
commutativité, 53
complexité, 46
décidable, 27
groupage, 53
immédiate, 26
indécidable, 27, 29, 45, 78, 79, 91
puissance, voir transformation

réseau, 20, 60, 107
rayon de voisinage, 22
minimal, 22
règle de transition
globale, 20
locale, 20, 23
restriction, 24, 25
relation
comparaison, 52
isomorphisme, 22
préordre, 22, 39, 45, 54, 72, 85
simulation, 52, 71, 85
sous-automate, 23, 31
représentant canonique, 82
représentation graphique, 52

temps réel, 88
transformation, 50
abstraite, 52
carrée, 42
composition, 57, 62, 83
généralisée, 71
géométrique, 57, 83
ouverture, 61
PCS, 62
PCST, 84
P̃CS, 68
P̃CS', 69
P̃CST, 85
P̃CST', 85
pré-image, 58, 84
spatiale, 50, 60, 68
sympathique, 58, 63, 84
temporelle, 50, 62
translation, 50, 62
transitoire, 29
translation, 25
élémentaire, 30
généralisée, 77
pure, 82
régulière, 50

universalité, 87
calcul, voir Turing
faible, 55
forte, 55
indécidabilité, 91
intrinsèque, 49, 77, 80, 88
minimisation, 90
prototype, 89
Turing, 49, 87, 88

voisinage, 20, 21
carré, 22
classique, 22
Moore, 22
vecteur, 21
von Neumann, 22