

**UNIVERSITÉ DE PROVENCE**

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE: E.D. 184

# THÈSE

présentée en vue d'obtenir le grade de

Docteur de l'Université de Provence

spécialité: Informatique

par

**Martin Delacourt**

Sous la direction de **Bruno Durand** et **Victor Poupet**

## **Automates cellulaires: dynamique directionnelle et asymptotique typique**

Thèse soutenue le 5 décembre 2011 devant le jury composé de :

Bruno DURAND	Université de Montpellier 2	Directeur de thèse
Enrico FORMENTI	Université de Nice Sophia Antipolis	Examineur
Anahí GAJARDO	Universidad de Concepción	Relectrice
Victor POUPET	Université de Provence	Directeur de thèse
Karl SCHLECHTA	Université de Provence	Examineur
Véronique TERRIER	Université de Caen	Relectrice



# Remerciements

Ayant mis un point final à la partie scientifique de ce document, l'écriture de cette dernière page m'est rendue aisée par la joie que m'apporte la possibilité d'associer à ce travail tous les gens qui ont aidé à sa réalisation. Si la crainte d'oublier quelqu'un qui ne se retrouverait que dans le remerciement générique final ne retiendra pas les doigts qui parcourent ce clavier, je tiens tout de même à m'excuser auprès de ces hypothétiques (pas tant que ça) personnes.

Pour commencer, je souhaite adresser un grand merci à Victor, mon directeur de thèse, qui a suivi de près mon travail, a toujours su poser les questions justes et m'a orienté tout en me laissant entièrement libre dans ma première expérience scientifique. Et puis, je voudrais remercier Bruno, mon directeur de thèse lui aussi, pour avoir accepté de l'être déjà, et pour tous ses conseils et suggestions scientifiques ou autres.

Merci aussi à Véronique et Anahí qui ont attentivement relu mon manuscrit, et à Enrico et Karl d'avoir accepté de siéger dans mon jury.

Toute l'équipe Escape, qui a aujourd'hui mérité son nom, a joué un rôle majeur au cours de ces années de doctorat : merci à Emmanuel pour sa culture scientifique et son remarquable choix de thésard, à Nicolas pour toutes les questions auxquelles je n'ai que rarement su répondre, à Grégory d'enseigner la persévérance : un jour, je comprendrai le forcing. Merci à Andrei de ne pas avoir gardé de dossier compromettant et à Alexander pour ces discussions toujours enrichissantes. Merci aux doctorants : en particulier à Gaétan parce que l'idée des compteurs a fait du chemin, à Alexis et Pascal parce qu'ils vont si bien ensemble, à Alex avec qui j'ai partagé mes directeurs de thèse, à Tarik qui a su quitter Metz et bonsoir à Fabien.

C'est aussi grâce au cadre du LIF que ma thèse s'est bien passée, en particulier Sylvie et Martine qui rendent tout facile, mais aussi tous les gens avec lesquels j'ai eu la chance de pouvoir discuter. Merci à mes coauteurs, Guillaume et Mathieu qui m'ont proposé des questions passionnantes, et à Laurent parce que la Sorbonne !

Je tiens à remercier aussi Guillaume et Jérémie qui ont presque fait partie d'Escape et tous les coincheurs de mes années marseillaises.

Comme la vie ne saurait être que scientifique et que je ne souhaite pas découper ma vie en deux, je veux aussi remercier Mathilde en premier lieu, ma famille, mes amis, qui m'ont toujours orienté vers une vie de découvertes et de curiosité.

Pour clôturer cette page ponctuée de 14 « merci », il me reste comme annoncé à

ii

remercier tous les gens que je n'ai pas mentionné plus haut mais que j'ai côtoyé pendant ces années.

# Table des matières

<b>Table des notations</b>	<b>vii</b>
<b>Introduction</b>	<b>ix</b>
<b>1 Définitions</b>	<b>1</b>
1.1 Dynamique symbolique . . . . .	1
1.1.1 Langages et configurations . . . . .	1
1.1.2 Topologie . . . . .	1
1.1.3 Mesures . . . . .	2
1.1.4 Sous-shift . . . . .	2
1.2 Automates cellulaires . . . . .	4
1.2.1 Définition . . . . .	4
1.2.2 Système dynamique . . . . .	6
1.2.3 Exemples . . . . .	8
1.2.4 Ensemble limite . . . . .	8
1.2.5 Dynamique . . . . .	11
1.3 Techniques de construction d'un automate cellulaire . . . . .	12
1.3.1 Signaux . . . . .	12
1.3.2 Simuler une machine . . . . .	15
1.3.3 Produit cartésien . . . . .	17
<b>2 Conséquences</b>	<b>19</b>
2.1 Premières propriétés . . . . .	19
2.1.1 Ensemble de conséquences . . . . .	19
2.1.2 Exemples . . . . .	20
2.1.3 Encadrer les conséquences . . . . .	21
2.1.4 Quels ensembles ? . . . . .	22
2.1.5 Conséquences et calculabilité . . . . .	24
2.1.6 Etats dans un ensemble de conséquences . . . . .	26

2.1.7	Ensembles finis . . . . .	28
2.2	Conséquences paraboliques . . . . .	29
2.3	Compteurs . . . . .	33
2.3.1	Principe . . . . .	33
2.3.2	Construction . . . . .	34
2.3.3	Collisions . . . . .	34
2.3.4	Utilisation . . . . .	37
2.3.5	Généralisation . . . . .	37
<b>3</b>	<b>Directions bloquantes</b> . . . . .	<b>41</b>
3.1	Définitions . . . . .	41
3.2	Dynamiques . . . . .	46
3.2.1	Équicontinuité . . . . .	46
3.2.2	Sensibilité et expansivité . . . . .	48
3.2.3	Classification . . . . .	49
3.3	Ensembles de directions . . . . .	50
3.3.1	Ordre sur les fonctions . . . . .	50
3.3.2	Ensemble de directions bloquantes . . . . .	50
3.3.3	Propriétés de $\mathbf{A}(\mathcal{A})$ . . . . .	51
3.3.4	Propriétés de $\mathbf{A}'(\mathcal{A})$ . . . . .	54
3.3.5	Caractérisation . . . . .	56
3.3.6	Directions d'expansivité . . . . .	57
3.4	Cas particuliers . . . . .	59
3.4.1	Variations non bornées . . . . .	59
3.4.2	Point d'équicontinuité de direction non périodique . . . . .	63
3.4.3	Automates réversibles . . . . .	64
3.5	Restriction aux droites . . . . .	65
3.5.1	Réels récursivement énumérables . . . . .	65
3.5.2	Bornes de $\tilde{\mathbf{A}}(\mathcal{A})$ . . . . .	66
3.6	Obtenir les bornes . . . . .	67
<b>4</b>	<b>Ensembles <math>\mu</math>-limite</b> . . . . .	<b>73</b>
4.1	Définitions . . . . .	73
4.1.1	Exemples . . . . .	75
4.1.2	Densités . . . . .	77
4.1.3	$\mu$ -nilpotence . . . . .	80
4.2	Mots bloquants . . . . .	81
4.3	Construction . . . . .	84
4.3.1	L'automate . . . . .	85
4.4	Sous-shifts atteignables . . . . .	96

4.4.1	Union de sous-shifts calculables . . . . .	96
4.4.2	Applications . . . . .	99
4.4.3	Vers une caractérisation? . . . . .	101
4.5	Théorème de Rice . . . . .	105
4.5.1	Propriétés d'ensembles $\mu$ -limite . . . . .	105
4.5.2	Engendrer une configuration générique . . . . .	106
4.5.3	Théorème de Rice . . . . .	107
4.6	Moyenne de Cesaro . . . . .	114
4.6.1	Exemple . . . . .	114
4.6.2	Définition . . . . .	116
4.6.3	Presque nilpotence . . . . .	116
	<b>Bilan et perspectives</b>	<b>117</b>





# Table des notations

$c_{[i,j]}$	Sous-mot de la configuration $c$ (2)
$[u]_k$	Cylindre (2)
$\sigma$	Shift ou décalage (2)
$\mathcal{L}(\Sigma)$	Langage d'un sous-shift (3)
$\mathcal{A}, \mathcal{B}$	Automate cellulaire (4)
$r_{\mathcal{A}}$	Rayon d'automate cellulaire (4)
$\delta_{\mathcal{A}}$	Règle locale d'automate cellulaire (4)
$\langle x, t \rangle$	Site (6)
$\Lambda(\mathcal{A})$	Ensemble limite (8)
$\mathfrak{C}(u)$	Ensemble de conséquences (20)
$\mathcal{F}$	Ensemble des fonctions $\mathbb{N} \rightarrow \mathbb{Z}$ (41)
$Bl^h(\mathcal{A})$	Ensemble des configurations contenant un mot bloquant (42)
$Eq^h(\mathcal{A})$	Ensemble des points d'équicontinuité (45)
$\mathcal{F}_b$	Ensemble des fonctions à variations bornées (45)
$\Delta_\alpha$	Droite discrète de pente $\alpha$ (50)
$\mathbf{A}(\mathcal{A})$	Ensemble des directions selon lesquelles il existe un mot bloquant (50)
$\mathbf{A}'(\mathcal{A})$	Ensemble des directions bloquantes (50)
$\mathcal{A}^t \mu([u])$	Mesure de l'ensemble des antécédents de $[u]$ (74)
$\text{Ant}^t(u)$	Ensemble des antécédents de $u$ à l'étape $t$ (74)
$L_\mu(\mathcal{A})$	Langage persistant ou $\mu$ -limite (74)
$\Lambda_\mu(\mathcal{A})$	Ensemble $\mu$ -limite (74)
$d_v(u)$	Densité d'un mot dans un autre (77)
$d_c(u)$	Densité d'un mot dans une configuration (78)
$c_{DB}$	Configuration de Bruijn (106)

# Introduction

Pour simuler des mécanismes d'« auto-reproduction », J. von Neumann [29] introduit les automates cellulaires dans les années 50. Les champs d'utilisation de ces objets se sont multipliés, dans des domaines comme la biologie ou la physique, où ils sont vus comme un modèle discret de systèmes complexes. En effet, un automate cellulaire est composé d'une infinité de cellules régulièrement disposées. Chaque cellule ne communique qu'avec ses voisins directs, et évolue en fonction de son propre état et de celui de ses voisines. L'évolution est discrète et synchrone, ce qui en fait un modèle simple à décrire. Des simulations de systèmes physiques [5] à l'aide d'automates cellulaires ont été effectuées, montrant ainsi la capacité des automates cellulaires à générer des comportements divers. L'exemple célèbre du jeu de la vie de J. Conway présenté par M. Gardner [15] notamment, présente des évolutions complexes et souvent peu comprises. Une question importante dans cette optique est celle de l'émergence ou de l'auto-organisation. On constate dans de nombreux systèmes complexes qu'en imposant des règles d'évolution locales, on peut obtenir des comportements qui reflètent une organisation globale. Les parallèles avec des systèmes naturels sont courants, comme par exemple la formation des cristaux ou l'organisation d'une fourmilière. Le comportement d'une fourmi étudiée en tant qu'individu est facile à décrire, et pourtant l'évolution globale de la fourmilière est bien plus complexe.

Concernant les automates cellulaires en particulier, l'émergence peut se traduire par l'apparition sur des entrées aléatoires de schémas typiques, que l'on peut voir comme des motifs réguliers, des phases, stables pour l'automate cellulaire et qui coexistent en étant séparés par des transitions de phases, qui sont des ruptures dans la régularité. On ne sait pas prédire un tel comportement pour un automate cellulaire, bien que ce cas se produise fréquemment. Ces automates cellulaires correspondent à une des classes de la première classification d'automates cellulaires proposée par S. Wolfram [41]. Cette classification concernait les automates cellulaires élémentaires, c'est à dire l'ensemble des 256 règles locales d'automates cellulaires « minimaux » en termes de voisinage et de nombres d'états différents que peut prendre une cellule. Bien qu'expérimental et arbitraire, ce tri des automates cellulaires montrait la présence de nombreux automates cellulaires présentant une forme d'émergence. D'autres clas-

sifications selon divers critères ont été proposées [8], recoupant parfois celle de S. Wolfram. Il existe aussi des propositions de formalisation de certains phénomènes liés comme les fonds et particules par N. Ollinger et G. Richard [32, 33] ou par M. Pivato [34, 35, 36].

On peut aussi voir les automates cellulaires comme des systèmes dynamiques, en regardant l'orbite d'une configuration initiale. G. A. Hedlund [17] a en effet donné une caractérisation des fonctions globales correspondant aux automates cellulaires permettant de les voir comme des systèmes dynamiques continus. On peut visualiser l'orbite d'une configuration en superposant toutes les images successives de cette configuration. On crée alors un diagramme espace-temps en ajoutant une dimension temporelle à la dimension d'une configuration. Un résultat marquant dans cette optique est l'équivalence entre injectivité et bijectivité obtenue en combinant les résultats de E. F. Moore et J. Myhill [27, 28]. D'autres questions de dynamiques ont été posées, comme par exemple celles de l'équicontinuité ou de l'expansivité. On cherche à savoir si des configurations proches le restent sous l'action répétée d'un automate cellulaire. Cette optique a été en particulier considérée par P. Kurka [22] ou par F. Blanchard et A. Maass [2] avec comme résultats la caractérisation des automates cellulaires équicontinus ou l'équivalence entre l'existence de points d'équicontinuité et celle d'un « mur » (un mot qui coupe l'espace et donc le diagramme espace-temps en deux parties qui ne communiquent pas). Cette approche a aussi donné lieu à une classification des automates cellulaires [22].

Il existe encore une approche différente des automates cellulaires, ils sont en effet rapidement devenus un modèle de calcul étudié, notamment grâce à A. R. Smith [38] qui prouve l'universalité Turing du modèle, c'est-à-dire la capacité de simuler une machine de Turing universelle. Ils rejoignent ainsi le domaine de la théorie du calcul. La notion de calcul, qui est fondamentale en informatique, repose sur l'équivalence admise (thèse de Church-Turing) entre tous les modèles de calcul « raisonnables ». Les fonctions récursives définies par A. Church [6] ou les machines de Turing d'A. M. Turing [39] ont donc la même capacité calculatoire que tous les autres modèles. Les automates cellulaires ont toutefois une place à part parmi ces modèles. Bien qu'il soit possible d'y implémenter un calcul séquentiel, les automates cellulaires sont fortement parallèles, le calcul est effectué en tous points par des machines qui ne disposent individuellement que d'un faible pouvoir calculatoire. Les automates cellulaires sont un modèle simple de calcul parallèle et c'est donc naturellement de la communication et de la diversité des calculs que provient leur richesse. De nombreux résultats importants ont été prouvés, et on peut signaler entre autres l'existence d'automates cellulaires intrinsèquement universels [31], c'est-à-dire capables de simuler le comportement de n'importe quel automate cellulaire.

Les automates cellulaires sont donc un lien entre ces différents points de vue : la richesse des comportements produits, la capacité calculatoire et les diverses propriétés dynamiques concourent à les mettre au centre de multiples thématiques, et donc à créer des ponts entre elles. Comprendre le fonctionnement des automates cellulaires est ainsi naturellement justifié par les multiples implications que peuvent avoir des résultats dans ce domaine.

Comme dit précédemment, l'émergence est une thématique riche et encore largement incomprise. On peut chercher à déterminer quelles règles locales entraînent une auto-organisation de l'espace, mais on peut s'intéresser aussi aux types d'auto-organisation possibles. C'est-à-dire trouver l'ensemble des motifs périodiques stables pour l'automate cellulaire et susceptibles d'apparaître dans l'évolution de la plupart des configurations. Une première approche inspirée par la théorie des systèmes dynamiques est de considérer l'ensemble des configurations que l'on peut observer arbitrairement tard : l'ensemble limite. Cet ensemble a été très étudié et en particulier J. Kari a montré [21] qu'on ne peut pas décider si l'ensemble limite d'un automate cellulaire ne contient qu'une seule configuration. Il a ensuite utilisé ce résultat pour montrer que toute propriété non triviale des ensembles limite est indécidable [20].

En 1990, M. Hurley introduit les « attracteurs » [18] qui sont une extension des ensembles limite. Un attracteur est un sous-ensemble de l'ensemble des configurations stable par l'automate cellulaire et qui apparaît arbitrairement tard dans l'évolution de l'automate cellulaire. Chacune de ces approches traite de l'ensemble des comportements asymptotiques, sans distinguer les plus fréquents des autres. Pour faire cette distinction, on peut utiliser une mesure de probabilité sur l'ensemble des configurations, ce que fait M. Hurley [19] en introduisant la notion de  $\mu$ -attracteur qui voit l'ensemble des configurations comme un ensemble mesurable. Cette piste aboutit à la définition de l'ensemble  $\mu$ -limite par P. Kůrka et A. Maass [23], qui est l'ensemble des configurations que l'on peut observer à la fois arbitrairement tard et souvent au sens de la mesure  $\mu$ .

Dans ce mémoire, nous étudierons les formes que peut prendre l'ensemble  $\mu$ -limite d'un automate cellulaire. Quelles contraintes doit respecter un ensemble pour pouvoir être un ensemble  $\mu$ -limite, quelle complexité peut-il atteindre ? Pour cela nous nous intéresserons précédemment au cas particulier des automates cellulaires pour lesquels il existe un mot bloquant, c'est-à-dire un mot qui, lorsqu'il est présent dans une configuration, la sépare en deux semi configurations dont les évolutions sont indépendantes. Un mot bloquant crée en effet au cours du temps un « mur » qui bloque toute communication entre les deux parties ainsi isolées. Ces murs sont notamment un moyen de créer des espaces indépendants au coeur des configurations.

En utilisant cette propriété, L. Boyer, V. Poupet et G. Theyssier [4] ont montré qu'on pouvait décrire l'ensemble  $\mu$ -limite à l'aide des mots apparaissant entre les murs. Nous utiliserons le point de vue de la dynamique directionnelle introduite par M. Sablik [37] qui s'intéresse à la dynamiques d'un automate cellulaire sous l'action d'un décalage. Un mur se visualise alors comme une rupture dans le diagramme espace-temps selon une direction qui dépend du décalage considéré. Nous généraliserons ce point de vue en regardant un ensemble plus large de décalages. Nous verrons donc dans un premier temps comment construire des automates cellulaires ayant des mot bloquants, et quelles sont leur propriétés.

## Organisation du mémoire

Dans une première partie, nous rappelons les définitions, notations ainsi que quelques techniques que nous utiliserons par la suite.

Le second chapitre est consacré à l'étude d'une notion préliminaire à celle de mot bloquant. Nous voyons comment la présence d'un mot fini dans la configuration initiale peut fixer l'état d'une partie des configurations images. Nous appelons cette partie l'ensemble de conséquences du mot. Après avoir explicité la définition, nous montrons quels types d'ensembles peuvent être des ensembles de conséquences, et nous montrons qu'on ne peut pas décider l'inclusion d'un sous ensemble simple dans l'ensemble des conséquences d'un mot. Puis nous donnons des exemples non triviaux, et en particulier celui d'un ensemble de conséquences ayant une frontière parabolique (dans une représentation spatio-temporelle). Pour finir ce chapitre, nous voyons une méthode générique pour construire des automates cellulaires et des mots ayant des ensembles de conséquences divers.

Le chapitre suivant traitera des mots dont l'ensemble de conséquences bloque la communication entre les deux demi-espaces qu'il sépare, c'est à dire des mots bloquants. Nous rappelons et généralisons les concepts de dynamique directionnelle dans le cas de directions quelconques, en particulier les liens entre équicontinuité et mot bloquant. Nous introduisons ensuite l'ensemble des directions selon lesquelles il existe un point d'équicontinuité pour un automate cellulaire donné. Nous voyons que cet ensemble est contraint et nous nous intéressons à deux restrictions : d'une part aux fonctions à variations bornées, d'autre part, aux fonctions linéaires. En particulier, dans ce dernier cas, nous donnons la caractérisation exacte des ensembles de directions linéaires selon lesquelles il existe un mot bloquant, ainsi qu'un exemple d'automate cellulaire associé à chaque ensemble.

Le dernier chapitre concerne les ensembles  $\mu$ -limite, c'est à dire l'ensemble des configurations témoignant uniquement des comportements asymptotiques fréquents. Dans un premier temps, nous nous intéressons aux automates cellulaires ayant un mot bloquant. Nous voyons que pour ces automates, on peut décrire exactement le langage de l'ensemble des configurations de l'ensemble  $\mu$ -limite. Si de plus, la direction bloquante n'est pas une droite de pente rationnelle, l'ensemble  $\mu$ -limite est restreint à un ensemble fini de configurations uniformes. Nous donnons ensuite une construction générique pour obtenir des automates cellulaires ayant un ensemble  $\mu$ -limite donné et nous nous intéressons aux diverses propriétés qui pourraient mener à une caractérisation des ensembles qui peuvent être des ensembles  $\mu$ -limite. Enfin, nous montrons un équivalent du théorème de Rice pour les ensembles  $\mu$ -limite : toute propriété non triviale sur ces ensembles est indécidable.



# Chapitre 1

## Définitions

### 1.1 Dynamique symbolique

#### 1.1.1 Langages et configurations

On commence par s'intéresser aux langages sur alphabets finis. On note  $X$  un ensemble fini que l'on appelle *alphabet*. On définit ensuite les *mots* sur cet alphabet qui sont les suites finies d'éléments de  $X$ . Un *langage* est un ensemble de mots sur un alphabet, donc en particulier  $X^* = \bigcup_{n \in \mathbb{N}} X^n = \{a_0 a_1 \dots a_n, n \in \mathbb{N}, \forall i \leq n, a_i \in X\}$  est le langage de tous les mots finis sur l'alphabet  $X$ . Et tout langage  $\mathcal{L}$  est un sous-ensemble de  $X^*$ .

On étend la définition de mot à des tailles infinies :

**Définition 1.1.1.** Une *configuration* sur l'alphabet  $X$  est un élément de  $X^{\mathbb{Z}}$ . Une *demi-configuration* sur l'alphabet  $X$  est un élément de  $X^{\mathbb{N}}$ .

Pour une configuration  $c \in X^{\mathbb{Z}}$  (ou une demi-configuration), on note  $c_{[i,j]}$  le mot contenu entre les indices  $i \in \mathbb{Z}$  et  $j \in \mathbb{Z}$  et on abrège  $c_{[i,i]}$  en  $c_i$  lorsque  $i = j$ . On appelle *langage* d'une configuration l'ensemble  $\mathcal{L}(c) = \{u \in X^*, \exists i, j \in \mathbb{Z}, c_{[i,j]} = u\}$ .

#### 1.1.2 Topologie

On définit une métrique sur  $X^{\mathbb{Z}}$  :

**Définition 1.1.2.** On utilisera la distance  $d$  sur  $X^{\mathbb{Z}}$  :

$$\forall x, y \in X^{\mathbb{Z}}, d(x, y) = 2^{-j} \text{ où } j = \min\{i \in \mathbb{Z}, x_i \neq y_i \text{ ou } x_{-i} \neq y_{-i}\}$$

Deux configurations sont donc proches lorsqu'elles coïncident sur un large segment en leur centre : pour  $x, y \in X^{\mathbb{Z}}$ ,  $d(x, y) < 2^{-i} \Leftrightarrow x_{[-i,i]} = y_{[-i,i]}$ . Pour la topologie associée, il existe une base naturelle d'ouverts-fermés :

**Définition 1.1.3.** Le *cylindre* associé au mot  $u \in X^*$  en position  $k \in \mathbb{Z}$  noté  $[u]_k$  est l'ensemble des configurations contenant  $u$  en position  $k$  :  $[u]_k = \{c \in X^{\mathbb{Z}}, c_{[k, k+|u|-1]} = u\}$ .

En particulier, on note  $[u] = [u]_{-\lfloor \frac{|u|-1}{2} \rfloor}$  le cylindre centré sur la position 0. Ces cylindres sont à la fois ouverts et fermés et pour les mots de taille impaire, ce sont les boules de la topologie induite par  $d$  : pour tout mot  $u \in X^*$ , si  $|u|$  est impair,  $[u]$  est la boule  $B(c, \varepsilon)$  de centre  $c \in [u]$  et de rayon  $\varepsilon = 2^{-\lfloor \frac{|u|-1}{2} \rfloor}$ .

Cette topologie rend  $X^{\mathbb{Z}}$  compact, en effet, en raison du nombre fini de cylindres associés à des mots de taille bornée, on peut extraire une suite convergente de toute suite dans  $X^{\mathbb{Z}}$  en fixant un mot central de plus en plus grand. Tout sous-ensemble fermé de  $X^{\mathbb{Z}}$  est donc lui aussi compact et en particulier tous les cylindres sont compacts.

La situation est semblable pour les demi-configurations, la distance associée correspond à l'indice de la première différence entre deux configurations.  $X^{\mathbb{N}}$  est lui aussi compact et les cylindres sont les ensembles de demi-configurations ayant un préfixe commun.

### 1.1.3 Mesures

On regardera l'espace des configurations  $X^{\mathbb{Z}}$  comme un espace mesurable. On peut utiliser comme tribu la famille des cylindres, et définir la mesure uniforme :

**Définition 1.1.4.** La mesure *uniforme*  $\mu$  est définie par :  $\forall u \in X^*, \forall k \in \mathbb{Z}, \mu([u]_k) = |X|^{-|u|}$ .

La distribution de probabilité dont elle est issue donne le même poids à tous les états de  $X$ , c'est la distribution non biaisée usuelle. Dans tout ce rapport, on note  $\mu$  cette mesure, et c'est la seule mesure à laquelle on s'intéressera.

La plupart des résultats concernant  $\mu$  peuvent être généralisés à des classes plus larges de mesures et en particulier les mesures de Bernoulli.

### 1.1.4 Sous-shift

On va définir une première dynamique sur  $X^{\mathbb{Z}}$ , en utilisant le décalage (ou shift)  $\sigma : X^{\mathbb{Z}} \rightarrow X^{\mathbb{Z}}$  tel que  $\forall c \in X^{\mathbb{Z}}, \forall i \in \mathbb{Z}, \sigma(c)_i = c_{i+1}$ . C'est une dynamique extrêmement simple qui se contente de déplacer la configuration d'une cellule vers la gauche. Clairement  $\sigma$  est continue et on peut maintenant voir  $(X^{\mathbb{Z}}, \sigma)$  comme un système dynamique. Il existe des sous-ensembles stables pour cette dynamique :

**Définition 1.1.5.** On appelle *sous-shift* ou *sous-décalage* un sous-ensemble fermé de  $X^{\mathbb{Z}}$  stable par  $\sigma$ .

Les sous-shifts sont donc des fermés inclus dans  $X^{\mathbb{Z}}$ , c'est-à-dire des compacts. On va donner quelques exemples de sous-shifts :  $X^{\mathbb{Z}}$  et  $\emptyset$  sont les sous-shifts les plus simples. Les sous-shifts finis sont des orbites (pour  $\sigma$ ) de configurations périodiques, ou des réunions finies de celles-ci. En effet, si un sous-shift contient une configuration non périodique, il contient nécessairement tous ses décalés qui sont en nombre infini.

On peut caractériser un sous-shift à l'aide du langage des mots qui apparaissent dans les configurations.

**Définition 1.1.6.** Pour  $\Sigma \subseteq X^{\mathbb{Z}}$ , on note

$$\mathcal{L}(\Sigma) = \{u \in X^*, \exists c \in \Sigma, \exists i, j \in \mathbb{Z}, c_{[i,j]} = u\}$$

le langage de  $\Sigma$ .

Lorsque  $\Sigma$  est restreint à une seule configuration  $c$ , on notera  $\mathcal{L}(c)$  le langage de  $\{c\}$ .

**Proposition 1.1.1.** Soit  $\Sigma_1$  et  $\Sigma_2$  des sous-shifts dans  $X^{\mathbb{Z}}$ ,  $\mathcal{L}(\Sigma_1) = \mathcal{L}(\Sigma_2)$  si et seulement si  $\Sigma_1 = \Sigma_2$ .

*Démonstration.* Clairement, si les sous-shifts sont égaux, leurs langages le sont aussi. On suppose donc  $\mathcal{L}(\Sigma_1) = \mathcal{L}(\Sigma_2)$ . Soit  $c \in \Sigma_1$ . Pour tout  $i \in \mathbb{N}$ , on note  $u_i = c_{[-i,i]}$ , alors  $u_i \in \mathcal{L}(\Sigma_1) = \mathcal{L}(\Sigma_2)$ , donc il existe  $c_i \in \Sigma_2$  telle que  $c_{i[-i,i]} = u_i$ . Comme  $\Sigma_2$  est compact, on peut extraire de la suite  $(c_i)_{i \in \mathbb{N}}$  une suite qui converge vers  $c' \in \Sigma_2$ .

Supposons que  $c' \neq c$ . Il existe  $j \in \mathbb{Z}$  tel que  $c'_j \neq c_j$ . Or il existe  $i_0 \in \mathbb{N}$ , tel que

$$\forall i \geq i_0, c'_{[-|j|,|j|]} = c_{i[-|j|,|j|]} = c_{[-|j|,|j|]}$$

Donc  $c' = c$  et  $c \in \Sigma_2$ .

Symétriquement, on a aussi  $\Sigma_2 \subseteq \Sigma_1$ . □

Un sous-shift est donc caractérisé par son langage ou, de manière équivalente, par le langage des mots qui n'apparaissent dans aucune configuration de ce sous-shift. Le langage d'un sous-shift est clairement stable par sous-mot.

On définit donc deux classes de sous-shifts que l'on rencontre fréquemment :

**Définition 1.1.7.** Un sous-shift est dit *de type fini* lorsqu'il peut être défini par un langage de mots interdits finis.

Un sous-shift est dit *sofique* lorsque son langage est rationnel.

Un sous-shift  $\Sigma$  est donc sofique lorsqu'il existe un graphe tel que les configurations de  $\Sigma$  sont exactement les chemins bi-infinis dans le graphe étiqueté par les arêtes.

Il est aussi possible de voir un sous-shift sofique  $\Sigma \subseteq X^{\mathbb{Z}}$  comme la projection d'un sous-shift de type fini  $\Sigma' \subseteq X'^{\mathbb{Z}}$  à l'aide de la fonction  $p : X' \mapsto X$  : si  $c \in \Sigma$ ,

alors il existe  $c' \in \Sigma'$  tel que  $\forall j \in \mathbb{Z}, c_j = p(c'_j)$ .

Par exemple, l'ensemble des configurations de  $\{0, 1\}^{\mathbb{Z}}$  qui ne contient pas deux 1 consécutifs. Le langage des mots interdits correspondant est  $\{11\}$ . Sur le même alphabet, l'ensemble des configurations qui ne contient qu'un seul 1 est sofique.

## 1.2 Automates cellulaires

On cherche maintenant à enrichir la dynamique sur  $X^{\mathbb{Z}}$ . On souhaite garder le caractère local et synchrone du décalage, mais autoriser des comportements plus diversifiés. On va donc donner une règle locale de transition que chaque cellule appliquera.

### 1.2.1 Définition

**Définition 1.2.1.** Un *automate cellulaire* (AC) est un triplet  $\mathcal{A} = (X, r, \delta)$  que l'on verra comme une fonction de  $X^{\mathbb{Z}}$  dans  $X^{\mathbb{Z}}$  avec :

- un ensemble fini d'états  $X$  appelé l'*alphabet* de  $\mathcal{A}$ ;
- un *rayon*  $r \in \mathbb{N}$ ;
- une *règle locale*  $\delta : X^{2r+1} \rightarrow X$ .

L'automate cellulaire est défini par :

$$\forall c \in X^{\mathbb{Z}}, \forall j \in \mathbb{Z}, \mathcal{A}(c)_j = \delta(c_{j-r}, c_{j-r+1}, \dots, c_{j+r})$$

Pour  $j \in \mathbb{Z}$ , l'emplacement  $j$  sera appelé la  $j^{\text{e}}$  cellule.

Un automate cellulaire est entièrement déterminé par sa règle locale. La définition de automates cellulaires, comme celle des sous-shifts, s'adapte de manière naturelle pour les dimensions supérieures, et de nombreuses études y sont aussi consacrées, mais on ne s'intéresse ici qu'aux automates cellulaires en dimension 1.

Dans la plupart des exemples que l'on verra par la suite,  $r$  sera égal à 1. En particulier,  $\sigma$  est un automate cellulaire de rayon 1, et l'identité (la règle locale  $\delta$  est l'identité) est un automate cellulaire de rayon 0.

Une caractérisation essentielle, qui montre le rôle prépondérant du décalage, a été montrée par G. A. Hedlund [17] :

**Théorème 1.2.1.** *Les automates cellulaires sur l'alphabet  $X$  sont exactement les fonctions continues de  $X^{\mathbb{Z}}$  dans  $X^{\mathbb{Z}}$  qui commutent avec  $\sigma$ .*

*Démonstration.* Un automate cellulaire est continu parce qu'il est local, et il commute avec le décalage.

Prenons maintenant une fonction  $f : X^{\mathbb{Z}} \mapsto X^{\mathbb{Z}}$  continue et telle que  $f \circ \sigma = \sigma \circ f$ . Comme  $X^{\mathbb{Z}}$  est compact et que  $f$  est continue,  $f$  est en fait uniformément continue, et donc il existe  $\eta > 0$  tel que :

$$\forall x, y \in X^{\mathbb{Z}}, d(x, y) < \eta \Rightarrow d(f(x), f(y)) < 1/2$$

On pose  $r = -\lceil \log_2 \eta \rceil$ . La propriété précédente devient :

$$\forall x, y \in X^{\mathbb{Z}}, x_{[-r, r]} = y_{[-r, r]} \Rightarrow f(x)_0 = f(y)_0$$

On a donc une règle locale  $\delta_0 : X^{2r+1} \mapsto X$  permettant de calculer l'image d'une configuration en la cellule 0. Il reste à vérifier que cette règle est identique pour chaque cellule.

Soit  $i \in \mathbb{Z}$ , pour toute configuration  $x \in X^{\mathbb{Z}}$  :

$$f(x)_i = (\sigma^i(f(x)))_0 = (f(\sigma^i(x)))_0 = \delta_0(\sigma^i(x)_{-r}, \dots, \sigma^i(x)_r) = \delta(x_{i-r}, \dots, x_{i+r})$$

Donc la règle locale  $\delta_0$  est bien la règle permettant de calculer l'image d'une configuration en chaque point. Et  $f$  est un automate cellulaire.  $\square$

Ce résultat montre notamment que les automates cellulaires sont très liés aux sous-shifts, en effet l'image d'un sous-shift par un automate cellulaire est un sous-shift, et comme  $X^{\mathbb{Z}}$  est un sous-shift, pour un automate cellulaire  $\mathcal{A}$ , et pour tout  $t \in \mathbb{N}$ , l'image  $\mathcal{A}^t(X^{\mathbb{Z}})$  est un sous-shift de  $X^{\mathbb{Z}}$ .

La fonction globale d'un automate cellulaire peut être injective, surjective ou bijective, et dans ce troisième cas, une deuxième conséquence du théorème précédent est que si un automate cellulaire est bijectif, alors son inverse est aussi une fonction continue qui commute avec  $\sigma$ , c'est à dire un automate cellulaire.

**Définition 1.2.2.** Un automate cellulaire  $\mathcal{A}$  est dit *réversible* lorsqu'il existe un automate cellulaire  $\mathcal{B}$  sur le même alphabet  $X$  tel que

$$\forall c \in X^{\mathbb{Z}}, \mathcal{B}(\mathcal{A}(c)) = \mathcal{A}(\mathcal{B}(c)) = c$$

**Proposition 1.2.2.** *Un automate cellulaire est réversible lorsque sa fonction globale est bijective.*

L'identité ou le décalage sont parmi les plus simples des automates réversibles.

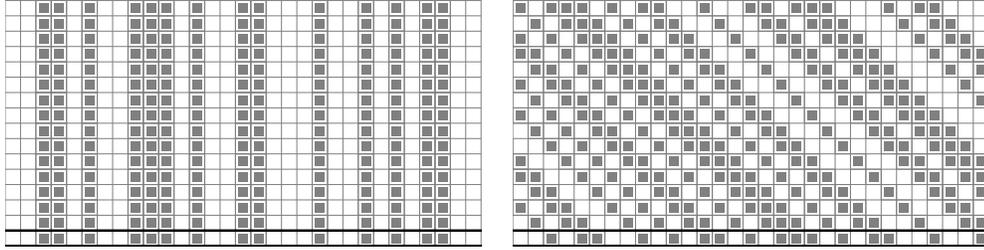


FIGURE 1.1 – Diagrammes espace-temps de l'identité et du décalage.

### 1.2.2 Système dynamique

Un automate cellulaire étant une fonction continue, on poursuit la généralisation du décalage  $\sigma$  vers un automate cellulaire quelconque en utilisant un automate cellulaire comme dynamique sur  $X^{\mathbb{Z}}$ . Pour un automate cellulaire  $\mathcal{A}$ ,  $(X^{\mathbb{Z}}, \mathcal{A})$  est un système dynamique et on diversifie ainsi les dynamiques engendrées.

Pour une configuration  $x \in X^{\mathbb{Z}}$ , on représentera l'orbite de  $x$  par son *diagramme espace-temps*, c'est à dire la superposition de toutes les images de  $x$  par l'automate cellulaire. Par exemple des diagrammes espace-temps de l'identité et de  $\sigma$  sont représentés en figure 1.1 avec l'alphabet  $\{\square, \blacksquare\}$ . Le temps évolue de bas en haut.

Lorsqu'un automate est réversible, le diagramme espace-temps peut se poursuivre aussi vers le bas.

**Définition 1.2.3.** On appellera *site* un emplacement du diagramme espace-temps, c'est à dire qu'un site est la donnée d'une cellule et d'un temps. On note  $\langle x, t \rangle \in \mathbb{Z} \times \mathbb{N}$  le site correspondant à la cellule  $x$  au temps  $t$ . Pour un automate réversible,  $t \in \mathbb{Z}$ .

L'état d'un site dépend de la configuration initiale, on peut donc parler des cellules qui déterminent l'état d'un site :

**Définition 1.2.4.** Soit  $\mathcal{A}$  un automate cellulaire de rayon  $r$ , les *antécédents* d'un site  $\langle x, t \rangle \in \mathbb{Z} \times \mathbb{N}$  sont les sites  $\{\langle x + j, t - 1 \rangle, -r \leq j \leq r\}$ .

Ainsi connaître l'état de tous les antécédents d'un site est suffisant pour déterminer l'état du site. On peut naturellement étendre la définition aux antécédents à l'étape  $n \in \mathbb{N}$  qui sont les sites  $\{\langle x + j, t - n \rangle, -rn \leq j \leq rn\}$ . La figure 1.2 présente les antécédents d'un site aux deux premières étapes pour un automate cellulaire de rayon 2.

On peut aussi faire un raisonnement inverse : si on connaît les états d'un ensemble  $S$  de sites, on connaît les états de toutes les cellules qui n'ont pour antécédents que des sites de  $S$ . En d'autres termes, on peut définir une notion d'image d'un mot fini par un automate cellulaire :

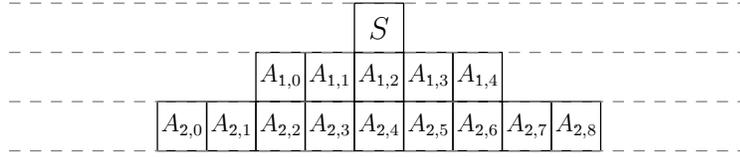


FIGURE 1.2 – Automate cellulaire de rayon 2 : les  $A_{1,i}$  sont les antécédents du site  $S$  à l'étape 1, et les  $A_{2,i}$  à l'étape 2.

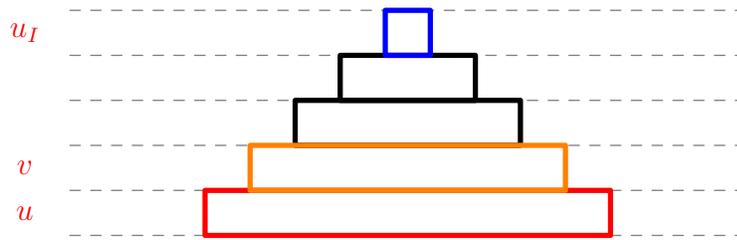


FIGURE 1.3 – Automate cellulaire de rayon 1 :  $v$  est l'image de  $u$  et  $u_I$  le résultat du calcul de  $\mathcal{A}$  sur  $u$ , quel que soit le contexte.

**Définition 1.2.5.** Soit  $\mathcal{A}$  un automate cellulaire de rayon  $r$ , et  $u \in X^*$  avec  $|u| \geq 2r + 1$ . On dit que  $v$  est l'image de  $u$  par  $\mathcal{A}$  lorsque :

- $|v| = |u| - 2r$
- $\forall 0 \leq j \leq |v| - 1, v_j = \delta(u_j, u_{j+1}, \dots, u_{j+2r+1})$

La figure 1.3 illustre cette définition. Si  $v$  est l'image de  $u$  par  $\mathcal{A}$ , on a donc  $\mathcal{A}([u]) \subseteq [v]$ . On peut ensuite étendre ce procédé tant que l'image obtenue est de taille supérieure à  $(2r + 1)$ .

**Définition 1.2.6.** Soit  $\mathcal{A}$  un automate cellulaire de rayon  $r$  et  $u \in X^*$ . On appelle calcul de  $\mathcal{A}$  sur  $u$  la famille  $(u_i)_{i \leq I}$  de mots telle que :

- $u_0 = u$  ;
- pour tout  $i$ ,  $u_{i+1}$  est l'image de  $u_i$  par  $\mathcal{A}$  ;
- $|u_I| \leq 2r$ .

Le mot  $u_I$  est le résultat de ce calcul.

On a alors  $\mathcal{A}^I([u]) \subseteq [u_I]$ . La figure 1.3 illustre cette notion.

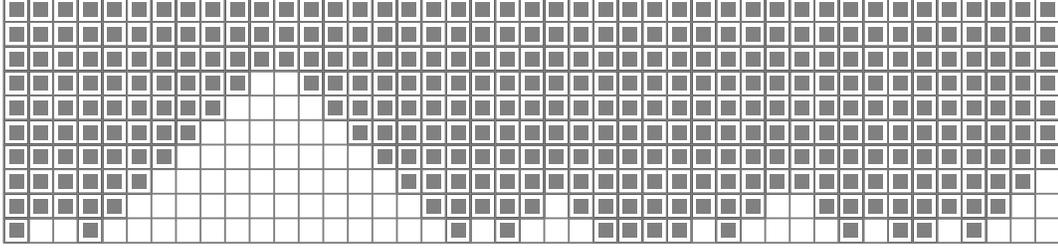


FIGURE 1.4 – Diagramme espace-temps de MAX.

### 1.2.3 Exemples

On présente ici deux nouveaux exemples simples d'automates cellulaires. On dit qu'un état  $q \in X$  est *quiescent* lorsque  $\delta(q, q, \dots, q) = q$ , c'est à dire qu'un site dont tous les antécédents sont dans l'état quiescent, l'est aussi. Par exemple, si  $q$  est quiescent, l'image de  $q^{\mathbb{Z}}$  est  $q^{\mathbb{Z}}$ .

On appelle *envahissant* un état  $q \in X$  tel que  $\delta(q_0, q_1, \dots, q_{2r}) = q$  dès qu'il existe  $0 \leq j \leq 2r$  tel que  $q_j = q$ . Si un site a parmi ses antécédents un site dans l'état envahissant, il prend lui aussi cet état.

On définit maintenant l'automate MAX sur l'alphabet  $\{\square, \blacksquare\}$  avec  $\blacksquare$  envahissant et  $\square$  quiescent. Le diagramme espace-temps de la figure 1.4 correspond à l'automate MAX. Un site est dans l'état  $\square$  lorsque ses antécédents sur la configuration initiale sont tous dans l'état  $\square$ .

On dit qu'un état  $q \in X$  est *persistant* si lorsque  $\mathcal{A}^t(c)_x = q$  pour  $c \in X^{\mathbb{Z}}$ ,  $(x, t) \in \mathbb{Z} \times \mathbb{N}$ , alors  $\mathcal{A}^{t+1}(c)_x = q$  aussi. Si une cellule est dans l'état  $q$  persistant, alors elle ne changera jamais plus d'état. Par exemple, tous les états de l'automate identité sont persistants. Un état envahissant est aussi persistant. Le diagramme de la figure 1.5 est celui d'un automate pour lequel l'état  $\blacksquare$  est persistant.

### 1.2.4 Ensemble limite

Un objet classique pour étudier les systèmes dynamiques en général et les automates cellulaires en particulier est l'*ensemble limite* qui est en quelque sorte l'image de  $X^{\mathbb{Z}}$  après un temps infini, et qui permet donc de s'affranchir de tous les comportements qui ne peuvent se produire que pendant les premières étapes.

**Définition 1.2.7.** Pour un automate cellulaire  $\mathcal{A}$ , on définit l'ensemble limite de  $\mathcal{A}$  comme étant

$$\Lambda(\mathcal{A}) = \bigcap_{t \in \mathbb{N}} \mathcal{A}^t(X^{\mathbb{Z}}) = \{c \in X^{\mathbb{Z}}, \forall t \in \mathbb{N}, \exists c_t \in X^{\mathbb{Z}}, \mathcal{A}^t(c_t) = c\}$$

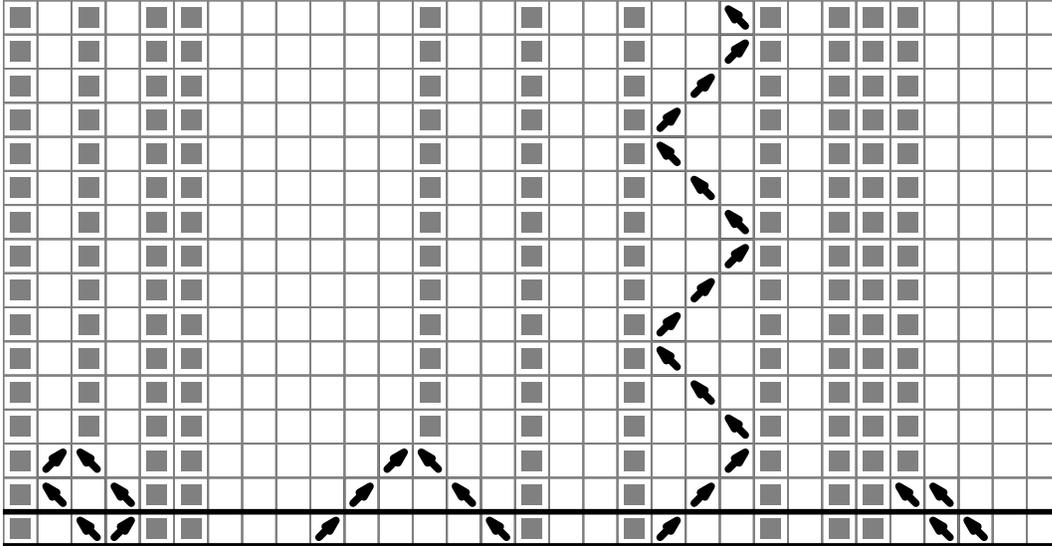


FIGURE 1.5 – Avec un état persistant.

C'est donc l'ensemble des configurations qui ont une préimage à tout temps. Par exemple pour un automate cellulaire réversible, l'ensemble limite est  $X^{\mathbb{Z}}$ . En fait on a l'équivalence suivante :

**Proposition 1.2.3.**  $\Lambda(\mathcal{A}) = X^{\mathbb{Z}} \Leftrightarrow \mathcal{A}$  est surjectif.

L'ensemble limite est particulièrement intéressant pour les automates cellulaires grâce à la propriété suivante :

**Proposition 1.2.4.** Pour un automate cellulaire  $\mathcal{A}$ ,  $\Lambda(\mathcal{A})$  est un sous-shift.

*Démonstration.* On commence par montrer que  $\Lambda(\mathcal{A})$  est invariant par décalage. Soit  $c \in \Lambda(\mathcal{A})$ , pour tout  $t \in \mathbb{N}$ , il existe  $c_t \in X^{\mathbb{Z}}$  telle que  $\mathcal{A}^t(c_t) = c$ . Donc  $\forall k \in \mathbb{Z}$ ,  $\mathcal{A}^t(\sigma^k(c_t)) = \sigma^k(c)$  et  $\sigma^k(c) \in \Lambda(\mathcal{A})$ .

Chaque itéré de  $\mathcal{A}$  est une fonction continue, donc l'image de  $X^{\mathbb{Z}}$  par un itéré de  $\mathcal{A}$  est fermé. L'intersection dénombrable des  $(\mathcal{A}^t(X^{\mathbb{Z}}))_t$  est fermée, donc  $\Lambda(\mathcal{A})$  est fermé.  $\square$

*Exemple 1.2.1.* On va calculer l'ensemble limite de MAX. On peut montrer que tous les motifs  $\square \blacksquare^k \square$  pour  $k \in \mathbb{N}$  ne peuvent apparaître après  $k/2$  étapes. En effet, la seule préimage de  $\square$  pour MAX est le mot  $\square^{2t+1}$  à l'étape  $t$ . Donc la préimage de  $\square \blacksquare^k \square$  à l'étape  $k/2 + 1$  devrait être dans  $\square^*$ . De même, on montre aisément que

toute configuration ne contenant aucun de ces motifs est dans l'ensemble limite de MAX qui contient donc toutes les configurations dans lesquelles il n'y a jamais de  $\blacksquare$  entre deux  $\square$  :

$$\Lambda(\text{MAX}) = (\mathbb{N}\blacksquare\square*\blacksquare\mathbb{N}) \cup (\square^{\mathbb{Z}}) \cup (\mathbb{N}\square\blacksquare\mathbb{N}) \cup (\mathbb{N}\blacksquare\square\mathbb{N})$$

On distingue certains automates cellulaires pour lesquels les images de toute configuration finissent par devenir uniformes :

**Définition 1.2.8.** Un automate cellulaire  $\mathcal{A}$  est dit *nilpotent* lorsqu'il existe  $s \in X$  tel que  $\Lambda(\mathcal{A}) = \{s^{\mathbb{Z}}\}$ .

Par exemple, si la règle locale donne toujours le même état en sortie, l'automate associé est nilpotent.

On peut montrer [7] que si un automate cellulaire  $\mathcal{A}$  est nilpotent, alors il existe  $t_0 \in \mathbb{N}$  tel que  $\forall t \geq t_0, \forall c \in X^{\mathbb{Z}}, \mathcal{A}^t(c) = s^{\mathbb{Z}}$ . Donc la configuration limite uniforme est atteinte en temps borné pour chaque automate. Toutefois, cette constante n'est pas calculable, on peut effectivement montrer [21] qu'on ne peut pas décider si un automate cellulaire est nilpotent ou non.

On peut aussi montrer la propriété suivante sur les ensembles limites :

**Proposition 1.2.5.** *Si l'ensemble limite d'un automate cellulaire est fini, alors l'automate est nilpotent.*

*Démonstration.* Supposons que l'automate  $\mathcal{A}$  ne soit pas nilpotent. Comme l'image d'une configuration uniforme est uniforme, il existe nécessairement un état  $s \in X$  et  $n \in \mathbb{N}$ , tels que  $\mathcal{A}^n(s^{\mathbb{Z}}) = s^{\mathbb{Z}}$ . Puisque  $\mathcal{A}$  n'est pas nilpotent, pour tout  $k \in \mathbb{N}$ , il existe un mot  $u \in X^{1+2rkn}$  tel que

$$\forall c \in [u], \mathcal{A}^{kn}(c) \neq s^{\mathbb{Z}}$$

En particulier  $\mathcal{A}^{kn}(\mathbb{N}sus^{\mathbb{N}}) \neq s^{\mathbb{Z}}$ , mais comme  $s$  est quiescent pour  $\mathcal{A}^n$ , il existe  $j \in \mathbb{N}$  tel que

$$c_k = \sigma^{-j} \left( \mathcal{A}^{kn}(\mathbb{N}sus^{\mathbb{N}}) \neq s^{\mathbb{Z}} \right) \in \mathbb{N}_s(X \setminus \{s\})X^{\mathbb{N}}$$

Comme  $X^{\mathbb{Z}}$  est compact, on peut extraire une suite convergente de la suite  $(c_k)_{k \in \mathbb{N}}$ , qui converge vers  $c \in \Lambda(\mathcal{A})$ . Et comme  $\forall k \in \mathbb{N}, c_k \in \mathbb{N}_s(X \setminus \{s\})X^{\mathbb{N}}$ , on a aussi  $c \in \mathbb{N}_s(X \setminus \{s\})X^{\mathbb{N}}$ .

La configuration  $c$  n'est pas périodique, et tous ses décalés sont dans l'ensemble limite de  $\mathcal{A}$  qui n'est donc pas fini.  $\square$

Enfin, Jarkko Kari a montré [20] un équivalent du théorème de Rice pour les ensembles limites d'automates cellulaires : toutes les propriétés non triviales des ensembles limite d'automates cellulaires sont indécidables.

### 1.2.5 Dynamique

On peut définir plusieurs types de dynamiques différents, en fonction des modes de propagation d'information dans l'automate [2, 22]. Le plus simple de ces modes de propagation intervient lorsqu'il n'y a qu'une seule direction de propagation :

**Définition 1.2.9.** Un automate cellulaire  $\mathcal{A}$  est dit *équicontinu* lorsque :  $\exists k \in \mathbb{N}, \forall c \in X^{\mathbb{Z}}$  :

- $\forall c' \in X^{\mathbb{Z}}$  tel que  $\forall i \leq k, c'_i = c_i$ , on a  $\forall t \in \mathbb{N}, \forall j \in \mathbb{N}, \mathcal{A}^t(c')_{-j} = \mathcal{A}^t(c)_{-j}$  ;
- $\forall c' \in X^{\mathbb{Z}}$  tel que  $\forall i \geq -k, c'_i = c_i$ , on a  $\forall t \in \mathbb{N}, \forall j \in \mathbb{N}, \mathcal{A}^t(c')_j = \mathcal{A}^t(c)_j$  ;

Ainsi pour chaque configuration, un changement d'un côté de l'origine ne se répercute pas de l'autre côté. En prenant des configurations décalées, on montre que cette propriété se vérifie en chaque cellule. Donc changer l'état d'une cellule d'une configuration ne peut changer au cours du temps que les cellules proches. On peut caractériser ces automates [22] :

**Proposition 1.2.6.** *Les automates cellulaires équicontinus sont exactement les automates ultimement périodiques, c'est à dire qu'il existe  $p, T \in \mathbb{N}$  tels que  $\forall t \geq p, \mathcal{A}^{t+T} = \mathcal{A}^t$ .*

Par exemple, les automates cellulaires nilpotents sont équicontinus, l'identité aussi.

Il existe une forme moins forte d'équicontinuité : lorsqu'un automate cellulaire possède des points d'équicontinuité :

**Définition 1.2.10.** Soit  $\mathcal{A}$  un automate cellulaire,  $c \in X^{\mathbb{Z}}$  est un point d'équicontinuité pour  $\mathcal{A}$  lorsque :

$$\forall \varepsilon, \exists \eta, \forall c' \in X^{\mathbb{Z}}, d(c, c') < \eta \Rightarrow \forall i \in \mathbb{N}, d(\mathcal{A}^i(c), \mathcal{A}^i(c')) < \varepsilon$$

Plus une configuration est proche d'un point d'équicontinuité, plus ses images successives sont proches des images du point d'équicontinuité. Il a été montré que l'existence d'un point d'équicontinuité est équivalente à l'existence d'un mot bloquant :

**Définition 1.2.11.** Un mot  $u \in x^*$  est dit *bloquant* pour  $\mathcal{A}$  lorsque pour toutes configurations  $c, c' \in [u]$  :

- si  $c_{]-\infty, 0]} = c'_{]-\infty, 0]}$  alors  $\forall i \in \mathbb{N}, \mathcal{A}^i(c)_{]-\infty, 0]} = \mathcal{A}^i(c')_{]-\infty, 0]}$  ;
- si  $c_{[0, \infty[} = c'_{[0, \infty[}$  alors  $\forall i \in \mathbb{N}, \mathcal{A}^i(c)_{[0, \infty[} = \mathcal{A}^i(c')_{[0, \infty[}$ .

C'est à dire qu'un mot bloquant sépare la configuration et empêche tout transfert d'information entre les deux demi-configurations qu'il sépare. Tout état persistant est

un mot bloquant dans un automate cellulaire de rayon 1 par exemple. Pour un automate cellulaire équicontinu, toutes les configurations sont des points d'équicontinuité. Donc tout mot fini assez grand est un mot bloquant.

On appelle parfois *presque équicontinu* un automate cellulaire qui possède des points d'équicontinuité mais qui n'est pas équicontinu. Un automate cellulaire qui ne possède pas de point d'équicontinuité est en revanche dit *sensible*.

Une dernière catégorie recouvre l'ensemble des automates cellulaires dans lesquels l'échange d'informations est généralisé : tout changement d'un état de la configuration initiale finit par se répercuter autour de n'importe quelle cellule.

**Définition 1.2.12.** Un automate cellulaire  $\mathcal{A}$  est dit expansif lorsque :

$$\exists \varepsilon > 0, \forall c, c' \in X^{\mathbb{Z}}, c \neq c' \Rightarrow \exists i \in \mathbb{N}, d(\mathcal{A}^i(c), \mathcal{A}^i(c')) > \varepsilon$$

### 1.3 Techniques de construction d'un automate cellulaire

Pour obtenir un automate cellulaire ayant un comportement spécifique, il faut définir la règle locale permettant d'obtenir le comportement global désiré. Pour cela, on utilisera quelques outils génériques permettant de faire du calcul et que l'on présente maintenant. Dans toutes les constructions que l'on réalisera, l'information sera portée par des états évoluant dans un fond uniforme et sans incidence sur ce qui le traverse. Pour faire cela, le fond sera un état quiescent que l'on note en général  $\square$ .

#### 1.3.1 Signaux

On cherche ici à tracer une demi-droite dans le diagramme espace-temps, c'est à dire à envoyer un signal à vitesse constante. Cela sera utilisé pour que les cellules communiquent entre elles, les signaux seront alors vus comme des informations que les cellules s'échangent. On se contente ici de vitesses rationnelles. Un signal sera constitué d'une succession d'états occupant une cellule à chaque étape. La figure 1.6 est l'illustration d'un signal de vitesse 1/2. L'état  $\blacksquare$  donne naissance sur la même cellule à un état  $\blacksquare$ , qui lui engendre un  $\blacksquare$  sur la cellule de droite. Ainsi l'alternance  $\blacksquare/\blacksquare$  produit un signal qui avance d'une cellule tous les deux temps.

La règle locale permettant de construire ce signal peut être la suivante :

- $\delta(s_1, \blacksquare, s_2) = \blacksquare$ ;
- $\delta(\blacksquare, s_1, s_2) = \blacksquare$ ;
- $\delta(s_1, s_2, s_3) = \square$  sinon.

Si deux signaux sont adjacents, on favorise la règle 1 :  $(\delta(\blacksquare, \blacksquare, s_2) = \blacksquare)$ , ce qui a pour effet de supprimer le signal de gauche. Il est possible de gérer la collision autrement.

Pour généraliser, on prend une vitesse rationnelle  $v = p/q \in \mathbb{Q}$ . On va alors utiliser  $q$  états  $a_0, a_1, \dots, a_{q-1}$  pour le signal, et pour tout  $t \in \mathbb{N}$ , l'image au temps  $t$

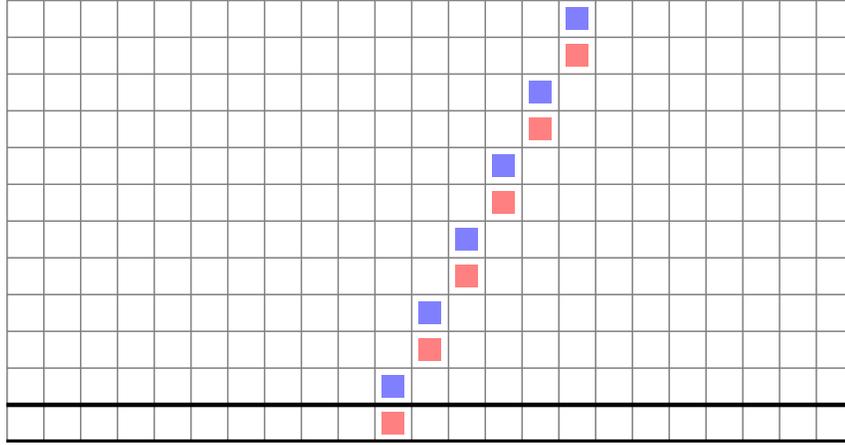


FIGURE 1.6 – Signal de vitesse 1/2.

de la configuration  $c = \mathbb{N} \square a_0 \square \mathbb{N}$  ne contient que l'état  $\square$  sauf en  $\lfloor vt \rfloor$  où  $\mathcal{A}^t(c)_{\lfloor vt \rfloor} = a_{\lfloor t \text{ mod } q \rfloor}$ . Donc, pour  $v \leq 1$ , la fonction locale pour  $0 \leq j < q$  contient les règles suivantes :

- si  $\lfloor v(t+1) \rfloor - \lfloor vt \rfloor = 0$ , alors  $\delta(\square, a_j, \square) = a_{j+1}$  ;
- si  $\lfloor v(t+1) \rfloor - \lfloor vt \rfloor = 1$ , alors  $\delta(a_j, \square, \square) = a_{j+1}$ .

Pour de plus grandes vitesses, il est nécessaire d'augmenter le rayon de l'automate. En particulier, pour un signal de vitesse  $v \in \mathbb{N}$ , il faut un automate cellulaire de rayon au moins  $v$ .

On construira souvent des automates cellulaires avec des signaux de vitesses différentes, les collisions sont alors fréquentes. On utilisera ces collisions pour calculer, c'est à dire que les signaux pourront interagir lors de ces collisions pour supprimer certains des signaux incidents ou créer de nouveaux signaux à partir du lieu de collision. On obtient des diagrammes espace-temps semblables à celui illustré en figure 1.7.

On voudra parfois représenter les signaux plus simplement, on se placera alors dans le plan continu et le diagramme de la figure 1.7 sera remplacé par celui de la figure 1.8.

Il y a une perte d'informations en passant à une représentation continue, puisque l'on perd l'épaisseur des signaux, toutefois, on utilisera cette illustration en prenant garde de ne pas représenter de schémas illégaux.

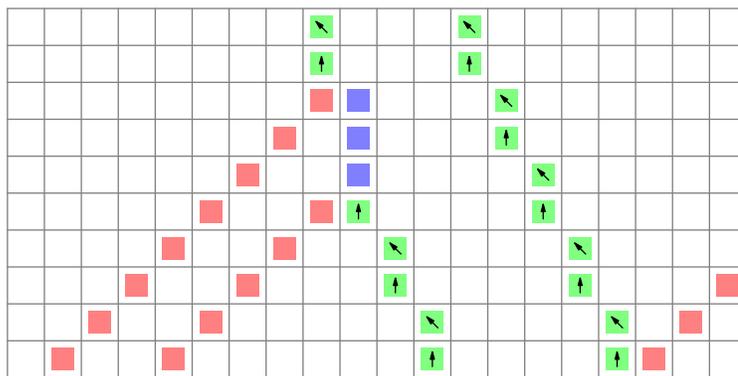


FIGURE 1.7 – Signaux et collisions.

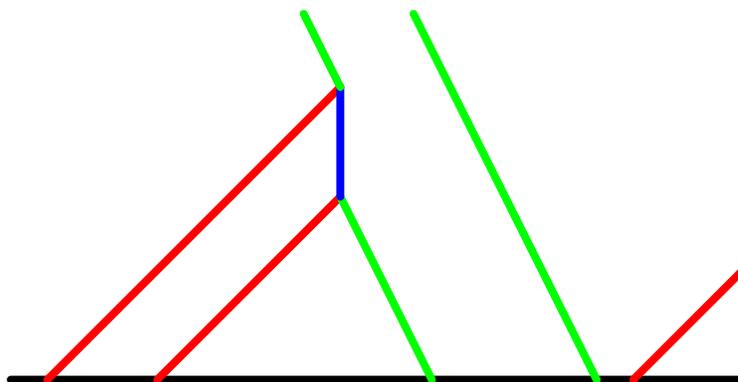


FIGURE 1.8 – Signaux et collisions :représentation continue.

■	1	$0^r$	1	0	1	0	1	1	1	■
■	$0^r$	1	1	0	1	0	1	1	1	■
■	1	1	1	0	1	0	1	1	1	■
■	$0^r$	0	1	0	1	0	1	1	1	■
■	1	$0^r$	0	0	1	0	1	1	1	■
■	$0^r$	1	0	0	1	0	1	1	1	■
■	1	1	0	0	1	0	1	1	1	■
■	0	1	0	0	1	0	1	1	1	■

FIGURE 1.9 – Simulation d'un compteur binaire.

### 1.3.2 Simuler une machine

On voit maintenant comment simuler le calcul d'une machine séquentielle dans un automate cellulaire au travers de deux exemples : les compteurs et les machines de Turing.

#### Compteurs

On commence par s'intéresser à la simulation de compteurs, on dispose donc d'un alphabet  $X$  contenant les états  $j \in [0, K - 1]$  pour un compteur en base  $K$ . On simule le fonctionnement du compteur en écrivant la valeur courante entre deux délimiteurs, par exemple ■. Puis la première cellule (sur la gauche) est incrémentée à chaque étape, et la gestion des retenues est gérée localement.

On illustre ce procédé sur un exemple : un compteur binaire en figure 1.9.

Les retenues sont propagées en parallèle et la valeur courante du compteur est connue modulo les retenues en transfert.

On peut combiner la simulation d'un compteur avec un décalage régulier, par exemple pour déplacer le compteur à une vitesse rationnelle  $v$ , chaque cellule du compteur constitue un signal se déplaçant à vitesse  $v$ . Il peut être nécessaire dans ce cas, d'augmenter le rayon pour pouvoir à la fois transmettre une retenue et avancer à vitesse  $v$ . Il est aussi possible d'avancer à une vitesse dictée par le déplacement

d'une des extrémités du compteur. Lorsque cette extrémité se déplace, tous les états composant le compteur suivent. Comme la règle est locale, le déplacement ne peut être simultané, il y aura nécessairement une latence, qui implique que des états se superposent ou que des cellules vides apparaissent dans le compteur.

### Machine de Turing

Les compteurs ne recouvrent qu'une très petite partie de l'ensemble des calculs que l'on peut être amené à effectuer. On va donc maintenant montrer comment simuler un modèle généraliste : les machines de Turing. Ainsi, tout algorithme pourra être implémenté. Pour les besoins de ce rapport, une simulation en espace fini est suffisante, on s'en contente donc ici.

La fonction de transition d'une machine de Turing est issue d'une règle locale comme pour un automate cellulaire, mais la transition n'est effectuée que sur la cellule où pointe la tête de calcul. Simuler un tel comportement dans un automate cellulaire ne nécessite donc que de donner un emplacement particulier contenant la tête et l'état de cette tête.

Pour une machine de Turing  $\mathcal{M}$  avec  $X_1$  comme alphabet de lecture/écriture,  $X_2$  comme alphabet interne et  $\phi$  comme fonction de transition, on prend un automate cellulaire tel que :

- l'alphabet  $X$  contient un délimiteur  $\blacksquare$ , et  $X_1$ , il contient aussi  $X_1 \times X_2$  qui correspond à l'ensemble des états du ruban de la machine où la tête de la machine, avec son état interne, pointe ;
- la règle locale de l'automate simule la règle de la machine : calcul du nouvel état interne, réécriture du ruban si nécessaire et déplacement de la tête.

Il suffit d'un automate cellulaire de rayon 1. La tête est bloquée lorsqu'elle rencontre un état  $\blacksquare$ , dans ce cas, la simulation est arrêtée. Ce fonctionnement est illustré par la figure 1.10.

On suppose ici que ces simulations ne se feront pas sur des configurations aléatoires, c'est à dire que l'on s'assurera qu'une simulation se fait avec une seule tête de calcul et sur un mot fini entre deux  $\blacksquare$ . Il est possible de gérer le cas de plusieurs têtes de calcul, soit en éliminant une partie d'entre elles, soit en attribuant une zone distincte à chaque tête. En revanche, sur un espace de calcul infini, il n'est pas possible d'assurer, en partant d'une configuration aléatoire qu'il existe une tête de calcul. On évite ici ces problèmes, puisque toutes les simulations seront maîtrisées et dans un environnement adapté.

■	0	1	0	1	1	0	1 <sup>q</sup>	1	1	■
■	0	1	0	1	1	0 <sup>q</sup>	1	1	1	■
■	0	1	0	1	1	0 <sup>q</sup>	1	1	1	■
■	0	1	0	1	0 <sup>q</sup>	0	1	1	1	■
■	0	1	0	1 <sup>q</sup>	1	0	1	1	1	■
■	0	1	0	1	1 <sup>q2</sup>	0	1	1	1	■
■	0	1	0	0 <sup>q1</sup>	1	0	1	1	1	■
■	0	1	0 <sup>q0</sup>	0	1	0	1	1	1	■

FIGURE 1.10 – Simulation d'une machine de Turing en espace fini.

### 1.3.3 Produit cartésien

Pour construire un automate cellulaire, il est souvent utilisé de décomposer le fonctionnement de l'automate en plusieurs évolutions simultanées, peu voire pas mutuellement dépendantes. Cela revient à utiliser plusieurs « couches » de calcul. On fait alors un *produit cartésien* d'automates cellulaires.

**Définition 1.3.1.** Soit  $\mathcal{A}$  et  $\mathcal{B}$  deux automates cellulaires sur les alphabets  $X$  et  $Y$ . Le *produit cartésien* de  $\mathcal{A}$  et  $\mathcal{B}$  est l'automate cellulaire  $\mathcal{D}$  sur l'alphabet  $X \times Y$  de rayon  $r_{\mathcal{D}} = \max(r_{\mathcal{A}}, r_{\mathcal{B}})$  et de fonction locale :

$$\delta_{\mathcal{D}}((x_0, y_0), \dots, (x_k, y_k)) = (\delta_{\mathcal{A}}(x_0, \dots, x_k), \delta_{\mathcal{B}}(y_0, \dots, y_k))$$

Cette définition stricte de produit cartésien de deux automates cellulaires permet de superposer les évolutions de deux automates cellulaires, mais pas de faire interagir les règles locales. Un tel automate n'apporte donc aucun nouveau comportement. Les constructions que l'on présente dans ce mémoire utiliseront une variante de ce principe. On autorise en effet des interactions entre les différentes « couches » de calcul.

On peut par exemple calculer sur une couche d'exécution tout en maintenant une couche de contrôle qui peut réinitialiser ou arrêter le calcul si nécessaire. Cela peut permettre la synchronisation de plusieurs calculs notamment. On peut aussi distinguer des couches d'écriture, de lecture, de calcul.



## Chapitre 2

# Conséquences

Dans ce chapitre, on cherche à forcer l'états de certains sites du diagramme espace-temps à l'aide d'un nombre fini de contraintes uniquement. On voudra notamment forcer l'état d'une infinité de sites, c'est à dire qu'en chacun de ce sites, tous les diagrammes espace-temps associés à des configurations initiales contenant un mot fini spécifique contiendront un même état. Donc une information finie permet de contraindre une infinité de sites et la communication entre les cellules ne doit favoriser cette information qu'en certains endroits.

On cherche en particulier à trouver de tels ensembles de sites ayant des formes non triviales. On montre que les états que l'on force ainsi ne peuvent pas être quelconques.

Les résultats de ce chapitre ont été publiés dans [10].

### 2.1 Premières propriétés

#### 2.1.1 Ensemble de conséquences

Une première solution pour fixer des états dans le diagramme espace-temps d'un automate cellulaire, est de choisir les états de tous les sites antécédents. Par exemple, pour fixer l'état du site  $\langle x, t \rangle$  avec un automate de rayon  $r$ , il suffit de choisir les états de tous les sites  $\langle x + k, 0 \rangle$  avec  $-rt \leq k \leq rt$ .

Cependant, si l'on s'intéresse à un ensemble infini de sites, l'ensemble des états antécédents est infini, en particulier, l'ensemble des sites  $\langle 0, t \rangle (t \in \mathbb{N})$  va demander de donner toute la configuration initiale. Cela ne présente pas d'intérêt, on va donc s'intéresser à ne fixer qu'un nombre fini de contraintes sur la configuration initiale. L'ensemble de tous les antécédents est nécessairement à proscrire, on va donc chercher des ensembles finis ayant la même propriété. On définit ainsi l'*ensemble de conséquences* d'un mot fini :

**Définition 2.1.1.** Soit  $\mathcal{A}$  un AC de rayon  $r$  et  $u \in X^*$  un mot fini sur son alphabet. On note :

$$E(u) = \left\{ \langle x, t \rangle \in \mathbb{Z} \times \mathbb{N}^*, -rt - \lfloor \frac{|u| - 1}{2} \rfloor \leq x \leq rt + \lfloor \frac{|u|}{2} \rfloor \right\}$$

L'ensemble de conséquences de  $u$  pour  $\mathcal{A}$  est :

$$\mathfrak{C}(u, \mathcal{A}) = \left\{ \langle x, t \rangle \in E(u), \forall c, c' \in [u], \mathcal{A}^t(c)_x = \mathcal{A}^t(c')_x \right\}$$

Lorsque le contexte n'est pas ambigu, on notera  $\mathfrak{C}(u)$ .

On peut se poser la question de la forme que peut prendre l'ensemble des conséquences d'un mot fini, sa taille, sa connexité. On verra aussi comment les états que peuvent prendre les sites de cet ensemble sont parfois contraints.

### 2.1.2 Exemples

On commence par quelques exemples pour illustrer la définition, les AC suivants n'ont que deux états :  $\blacksquare$  et  $\square$ ,  $\square$  correspond à un site qui peut être dans l'un ou l'autre des deux états selon la configuration initiale :

Avec l'AC identité de la figure 2.1, on peut obtenir une droite comme ensemble de conséquences :  $\mathfrak{C}(\blacksquare, \mathcal{A}) = \{ \langle 0, j \rangle, j \in \mathbb{N} \}$ , ou de manière plus générale autant de droites verticales que de cellules dont l'état est fixé. En prenant l'automate décalage, on obtient de la même manière des diagonales comme illustré en figure 2.2.

Le cône plein est un ensemble de conséquences pour l'AC avec un état envahissant de la figure 2.3 :  $\mathfrak{C}(\blacksquare, \mathcal{A}) = \{ \langle i, j \rangle, j \leq |i|, i \in \mathbb{Z}, j \in \mathbb{N}^* \}$ , et le damier pour l'AC de la figure 2.4.

En effet, dans ces quatre cas, on voit sur les illustrations qu'un état  $\blacksquare$  implique que tous les sites de l'ensemble de conséquences soient dans cet état  $\blacksquare$ . En revanche pour tout autre site, on peut en changeant la configuration initiale autour de l'état  $\blacksquare$  central, donner l'état  $\square$  ou l'état  $\square$  à ce site.

En s'autorisant plus d'états et en augmentant le rayon si nécessaire, on peut aussi obtenir des ensembles de conséquences suivant n'importe quel signal de vitesse rationnelle. La figure 2.5 l'illustre avec la vitesse  $\frac{1}{2}$ . L'idée est de construire un signal de vitesse  $\frac{1}{2}$  comme en 1.3.1. L'état  $\uparrow$  dans le site  $\langle i, j \rangle$  force l'état  $\nearrow$  dans le site  $\langle i, j + 1 \rangle$ , qui a son tour force l'état  $\uparrow$  dans le site  $\langle i + 1, j + 1 \rangle$ . Ainsi le mot  $\square \uparrow$  a pour conséquences l'ensemble

$$\{ \langle i, j \rangle, j = 2i \text{ ou } j = 2i + 1, i \in \mathbb{N}, j \in \mathbb{N} \}$$

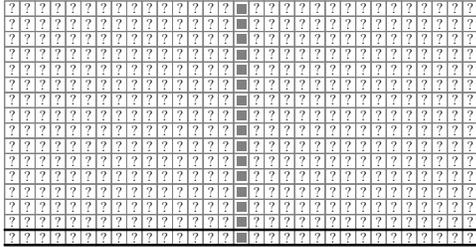


FIGURE 2.1 – Automate identité.

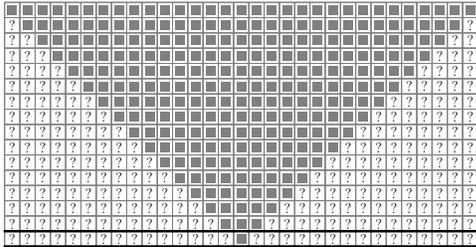


FIGURE 2.3 – AC avec état envahissant.

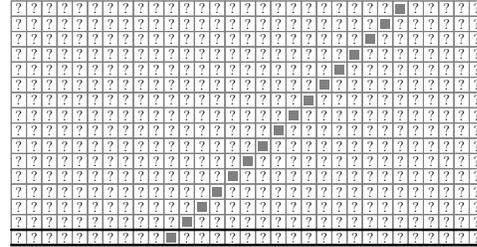


FIGURE 2.2 – Automate décalage.

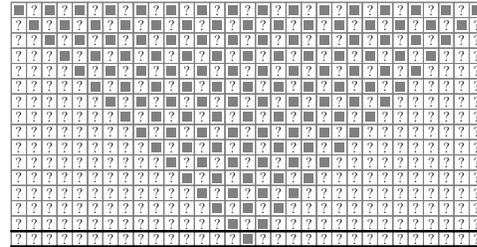


FIGURE 2.4 – AC où un état se propage uniquement à droite et à gauche.

### 2.1.3 Encadrer les conséquences

On a vu grâce aux exemples précédents, qu'un ensemble de conséquences peut prendre des formes diverses, on va voir ici que l'on peut restreindre l'ensemble des formes qui peuvent être ainsi atteintes.

En particulier, pour un automate de rayon  $r$ , un mot  $u$  de taille  $2r + 1$  a nécessairement un ensemble de conséquences non vide : un site a comme antécédents les sites de  $u$  exactement, donc son état est fixé. Plus généralement, si tous les antécédents d'un site sont dans l'ensemble de conséquences d'un mot, alors ce site est aussi dans l'ensemble de conséquences.

On prend un ensemble  $S \in \mathbb{Z} \times \mathbb{N}$  de sites. On construit alors un ensemble  $S' \in \mathbb{Z} \times \mathbb{N}$  minimal tel que :

- $S \subseteq S'$  ;
- si tous les antécédents d'un site  $s$  sont dans  $S'$ , alors  $s \in S'$  aussi.

$S'$  est le plus petit ensemble contenant  $S$  et potentiellement ensemble de conséquences.

Inversement, un site  $\langle x, t \rangle$  ne peut être dans les conséquences d'un mot  $u$  que s'il a des antécédents à l'étape 0 entre  $\langle -\lfloor \frac{|u|-1}{2} \rfloor, 0 \rangle$  et  $\langle \lfloor \frac{|u|}{2} \rfloor, 0 \rangle$ . On a pourtant requis

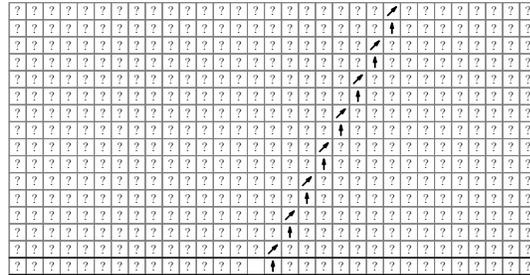


FIGURE 2.5 –

dans la définition que  $-rt - \lfloor \frac{|u|-1}{2} \rfloor \leq x \leq \lfloor \frac{|u|}{2} \rfloor + rt$ . Il peut en effet se produire que l'état d'un site soit forcé bien qu'aucun de ses antécédents à l'étape initiale ne soit dans  $u$ . Cela signifie que quelle que soit la configuration initiale, l'état de ce site est déterminé, et l'automate est donc nilpotent. Cette précision dans la définition sert donc uniquement à traiter le cas des AC nilpotents pour lesquels, à partir d'un certain temps, toute la configuration est dans l'ensemble de conséquences de n'importe quel mot.

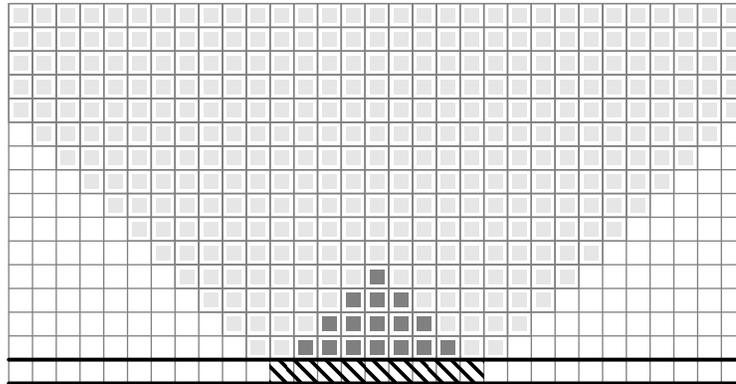


FIGURE 2.6 – Les conséquences du mot  $u$  (hachuré) sont contenues dans le cône gris clair et contiennent  $T^*(u)$  en gris foncé.

### 2.1.4 Quels ensembles ?

L'ensemble des ensembles de conséquences atteignables possède lui même une structure, en effet, la proposition suivante servira dans la suite à créer des ensembles

de conséquences spécifiques. Pour la prouver, on utilisera ce lemme :

**Lemme 2.1.1.** *Pour tous AC  $\mathcal{A}$  et mot  $u$  sur son alphabet, il existe un AC  $\mathcal{A}'$  et un mot  $u'$  sur son alphabet de taille  $|u'| = |u| + 1$  tel que  $\mathfrak{C}(u', \mathcal{A}') = \mathfrak{C}(u, \mathcal{A})$ .*

*Démonstration.* Notons  $\mathcal{A} = (X = \{a_0, a_1, \dots, a_{k-1}\}, r, \delta)$ , on construit  $\mathcal{A}' = (X' = X^k \times \{0, 1, S\}, r' = r + 1, \delta')$ . Une configuration est donc un  $k + 1$ -uplet contenant  $k$  configurations de  $X^{\mathbb{Z}}$  et un élément de  $\{0, 1, S\}^{\mathbb{Z}}$ . On note

$$u' = (ua_0, ua_1, \dots, ua_{k-1}, S^{|u|+1})$$

. La règle locale  $\delta'$  consiste à appliquer la règle  $\delta$  sur chacune des  $k$  premières composantes. Si ces  $k$  premières composantes contiennent le même état, on écrit  $S$  dans la dernière. Si ce n'est pas le cas, on regarde les dernières composantes des antécédents du site. Si l'une d'entre elles est à 1, on marque 1, sinon, si une est à 0, on marque 0 et sinon  $S$ .

L'idée est de calculer les images successives correspondant à toutes les configurations initiales contenant  $u$ . Comme on fixe un mot de taille  $|u| + 1$ , on prend toutes les configurations  $us, s \in X$ . C'est le rôle des  $k$  premières composantes. La dernière est un témoin qui indique si les premières sont toutes égales ou non. Ainsi si un site est dans les conséquences de  $u$  pour  $\mathcal{A}$ , les  $k$  calculs donneront le même état et la dernière composante du site sera  $S$ . Sinon, on pourra avoir 0 ou 1 sur la dernière composante.

Plus formellement, si  $\langle x, t \rangle \in \mathfrak{C}(u, \mathcal{A})$  pour  $x \in \mathbb{Z}, t \in \mathbb{N}^*$ , il existe  $s \in X$  tel que  $\forall c \in [u], \mathcal{A}^t(c)_x = s$ . C'est vrai en particulier pour toute configuration de  $[ua_i]$  pour  $i \in [0, k - 1]$ . Soit  $c = (c_0, c_1, \dots, c_{k-1}, c')$  une configuration de  $X'^{\mathbb{Z}}$ . On a  $\forall i \leq k - 1, \mathcal{A}^t(c_i)_x = s$ , et comme les  $(k - 1)$  premières composantes sont égales, on marque  $S$  sur la dernière donc  $\mathcal{A}^t(c)_x = (s, s, \dots, S)$ . Donc  $\langle x, t \rangle \in \mathfrak{C}(u', \mathcal{A}')$ .

Réciproquement, si  $\langle x, t \rangle \notin \mathfrak{C}(u, \mathcal{A})$  pour  $x \in \mathbb{Z}, t \in \mathbb{N}^*$ , on va montrer qu'il existe  $c'_0, c'_1 \in [u']$  telles que les dernières composantes de  $\mathcal{A}^t(c'_0)_x$  et  $\mathcal{A}^t(c'_1)_x$  soient respectivement 0 et 1. Il existe  $i, j \leq k - 1$  et  $c_1 \in [ua_i], c_2 \in [ua_j]$  telles que  $\mathcal{A}^t(c_1)_x \neq \mathcal{A}^t(c_2)_x$ . On définit  $c'_1$  pour que la  $i^e$  composante soit  $c_1$  et la  $j^e$   $c_2$ . La dernière composante sera  ${}^{\mathbb{N}}0S^k0^{\mathbb{N}}$ . Les autres sont quelconques dans  $[ua_l]$  pour chaque  $l \neq i, j$ .  $c'_2$  est identique à  $c'_1$  sauf sur la dernière composante qui est  ${}^{\mathbb{N}}1S^k1^{\mathbb{N}}$ .

On note  $E \subseteq \mathbb{Z} \times \mathbb{N}$  l'ensemble des sites où les composantes  $i$  et  $j$  ne sont pas égales. Cet ensemble est le même pour  $c'_0$  et  $c'_1$ . Considérons un site  $\langle y, \tau \rangle$  de  $E$ . Il a nécessairement un antécédent dans  $E$ , puisque la règle de transition est la même sur les deux composantes. On peut ainsi obtenir une suite d'antécédents dans  $E$  jusqu'au temps 0. Avec la configuration initiale  $c'_0$ , Aucun état n'a 1 sur sa dernière composante, donc cette propriété est vraie pour tout site.

On montre alors par récurrence sur le temps  $\tau \in \mathbb{N}$  que pour tout site  $\langle y, \tau \rangle$  de  $E$ , la dernière composante de l'état contient 0. Si  $\tau = 1$ , il existe un antécédent  $\langle y', 0 \rangle$

dans  $E$ . Si  $y' \neq |u'| - 1$ , la dernière composante de  $(c'_0)_{y'}$  contient 0 donc  $\mathcal{A}'(c'_0)_y$  aussi. Sinon,  $y' = |u'| - 1$ , et comme  $r' = r + 1$ ,  $\langle y, \tau \rangle$  a pour antécédent  $\langle |u'|, 0 \rangle$  qui a 0 sur sa dernière composante. Si  $\tau > 1$ , il existe un antécédent  $\langle y', \tau - 1 \rangle$  dans  $E$ . Et la dernière composante de  $\mathcal{A}'^{\tau-1}(c'_0)_{y'}$  vaut 0, ce qui permet encore de conclure.

On peut montrer la proposition symétrique avec  $c'_1$  en configuration initiale. On a donc bien montré que les dernières composantes de  $\mathcal{A}'^t(c'_0)_x$  et  $\mathcal{A}'^t(c'_1)_x$  sont respectivement 0 et 1. Il s'ensuit que  $\langle x, t \rangle \notin \mathfrak{C}(u', \mathcal{A}')$ .  $\square$

Ce lemme permet maintenant de démontrer que l'ensemble des ensembles de conséquences est stable par intersection :

**Proposition 2.1.2.** *Soient  $u_1$  et  $u_2$  deux mots sur les alphabets des AC  $\mathcal{A}_1$  et  $\mathcal{A}_2$ . Il existe un AC  $\mathcal{A}$  et un mot  $u$  sur son alphabet tels que  $\mathfrak{C}(u, \mathcal{A}) = \mathfrak{C}(u_1, \mathcal{A}_1) \cap \mathfrak{C}(u_2, \mathcal{A}_2)$ .*

*Démonstration.* Supposons que  $|u_1| \neq |u_2|$ , par exemple  $|u_1| < |u_2|$ , en utilisant le lemme 2.1.1 plusieurs fois si nécessaire, on obtient un automate  $\mathcal{A}'_1$  et un mot  $u'_1$  sur son alphabet tels que  $\mathfrak{C}(u'_1, \mathcal{A}'_1) = \mathfrak{C}(u_1, \mathcal{A}_1)$  et  $|u'_1| = |u_2|$ . On note  $\mathcal{A}'_1 = (X'_1, r'_1, \delta'_1)$  et  $\mathcal{A}_2 = (X_2, r_2, \delta_2)$  les deux AC. On construit alors l'AC  $\mathcal{A} = (X'_1 \times X_2, r = \max(r'_1, r_2), \delta)$  où  $\delta$  consiste à appliquer  $\delta'_1$  sur la première composante et  $\delta_2$  sur la seconde. En particulier, si  $c_1 \in X'^*_1$  et  $c_2 \in X^*_2$ ,  $\mathcal{A}(c_1, c_2) = (\mathcal{A}'_1(c_1), \mathcal{A}'_2(c_2))$ . Et on note  $u = (u'_1, u_2)$ .

On montre maintenant que  $\mathfrak{C}(u, \mathcal{A}) = \mathfrak{C}(u'_1, \mathcal{A}'_1) \cap \mathfrak{C}(u_2, \mathcal{A}_2)$ . Soit  $\langle x, t \rangle \in \mathfrak{C}(u, \mathcal{A})$ .  $\exists (s_1, s_2) \in X'_1 \times X_2, \forall c \in [u], \mathcal{A}^t(c)_x = (s_1, s_2)$ . Mais  $c = (c_1, c_2) \in X'^{\mathbb{Z}}_1 \times X^{\mathbb{Z}}_2$ , donc  $\mathcal{A}^t(c)_x = (\mathcal{A}'^t_1(c_1)_x, \mathcal{A}^t_2(c_2)_x)$ . Il s'en suit que  $\forall c_1 \in [u'_1], \mathcal{A}'^t_1(c_1)_x = (s_1)$  et  $\forall c_2 \in [u_2], \mathcal{A}^t_2(c_2)_x = (s_2)$ . Ce qui signifie que  $\langle x, t \rangle \in \mathfrak{C}(u'_1, \mathcal{A}'_1)$  et  $\langle x, t \rangle \in \mathfrak{C}(u_2, \mathcal{A}_2)$ .

Soit maintenant  $\langle x, t \rangle \in \mathfrak{C}(u'_1, \mathcal{A}'_1) \cap \mathfrak{C}(u_2, \mathcal{A}_2)$ . Il existe donc  $s_1 \in X'_1$  et  $s_2 \in X_2$  tels que  $\forall c_1 \in X'^{\mathbb{Z}}_1, \mathcal{A}'^t_1(c_1)_x = s_1$  et  $\forall c_2 \in X^{\mathbb{Z}}_2, \mathcal{A}^t_2(c_2)_x = s_2$ . Prenons une configuration  $c = (c_1, c_2) \in X'^{\mathbb{Z}}_1 \times X^{\mathbb{Z}}_2$ , alors  $\mathcal{A}^t(c)_x = (\mathcal{A}'^t_1(c_1)_x, \mathcal{A}^t_2(c_2)_x) = (s_1, s_2)$ . Finalement  $\langle x, t \rangle \in \mathfrak{C}(u, \mathcal{A})$ .  $\square$

### 2.1.5 Conséquences et calculabilité

**Proposition 2.1.3.** *Pour un mot  $u$  sur un AC  $\mathcal{A}$ ,  $\mathfrak{C}(u, \mathcal{A})$  est calculable.*

*Démonstration.* Soit  $r$  le rayon de  $\mathcal{A}$ . Pour une configuration donnée et un site  $\langle x, t \rangle \in \mathbb{Z} \times \mathbb{N}$ , calculer l'état de  $\langle x, t \rangle$  revient à calculer le « triangle » au dessus de ce mot, c'est à dire les antécédents successifs du site. C'est donc un calcul fini.

En se donnant un mot  $u$ , on ne s'intéresse plus qu'à l'ensemble des configurations contenant ce mot  $u$  en position 0. Pour un site donné, il suffit de regarder tous les contenus possibles des antécédents de ce site à l'étape 0. Donc un nombre fini de calculs à effectuer. On peut donc calculer tous les états qui peuvent être pris par ce site et ainsi déterminer s'il est dans les conséquences ou non.  $\square$

Par exemple, si on se donne un ensemble d'entiers non calculables  $A$ , l'ensemble  $\{\langle 0, x \rangle, x \in A\}$  ne peut pas être un ensemble de conséquences. Toutefois si l'ensemble est calculable, on va montrer qu'il est en revanche indécidable de savoir si une forme simple  $y$  est incluse : par exemple une colonne.

**Proposition 2.1.4.** *Il n'existe pas d'algorithme prenant en entrée un AC et un mot  $u$  sur son alphabet et décidant si  $\{\langle 0, t \rangle, t \in \mathbb{N}\} \subseteq \mathfrak{C}(u)$ .*

*Démonstration.* L'idée est de réduire ce problème à celui de l'arrêt des machines de Turing. Pour cela, le mot  $u$  sera un espace de calcul dans lequel une machine va évoluer, et sur sa gauche, la colonne d'abscisse 0 sera le témoin de cet espace. On s'assurera que cet espace ne puisse être modifié depuis l'extérieur, en revanche, si la machine s'arrête, il sera détruit, et l'état sur la cellule origine sera changé. Ainsi, si la machine ne s'arrête pas, la cellule d'origine ne changera jamais d'état et  $\{\langle 0, t \rangle, t \in \mathbb{N}\} \subseteq \mathfrak{C}(u)$ . On veillera ensuite à ce que l'espace de calcul alloué à cette machine puisse s'accroître pour que le calcul soit effectué.

Formellement, l'arrêt d'une machine de Turing étant indécidable avec un nombre borné d'états et de symboles, on utilise un alphabet  $X_1$  permettant de simuler une machine de Turing. On ajoute deux états  $\blacksquare$  et  $\boxtimes$  qui seront respectivement les bords gauche et droit de l'espace de calcul et un état quiescent :  $X = X_1 \cup \{\boxtimes, \blacksquare, \square\}$ . On définit maintenant  $u$  comme étant successivement un état  $\blacksquare$ , un mot de  $X_1^*$  qui sera le codage d'une entrée pour la machine de Turing avec une tête de calcul sur la première cellule, puis un état  $\boxtimes$ .

On va associer un AC  $\mathcal{A}$  à chaque machine de Turing. L'état  $\blacksquare$  ne peut être détruit que par une machine venant de la droite. L'état  $\boxtimes$ , lui, ne peut pas être détruit mais déplacé par une machine venant de la gauche. La simulation du calcul de la machine de Turing se fait normalement entre  $\blacksquare$  et  $\boxtimes$  jusqu'à ce que l'espace soit trop petit pour continuer le calcul. La machine regarde alors le bord droit de son espace de calcul, si  $\boxtimes$  est suivi par un  $\blacksquare$ , elle attend jusqu'à ce que ce ne soit plus le cas. Si  $\boxtimes$  n'est pas suivi par  $\blacksquare$ , elle déplace  $\boxtimes$  d'une cellule vers la droite et agrandit ainsi son espace de calcul.

Si le calcul effectué par la machine s'arrête, elle détruit les états  $\boxtimes$  et  $\blacksquare$  autour de son espace de calcul et les remplace par  $\square$ .

Prenons maintenant une machine  $\mathcal{M}$  et son AC associé  $\mathcal{A}$ . Pour un mot  $v$  d'entrée de  $\mathcal{M}$ , on considère le mot  $u = \blacksquare^{|v'|+2}v'\boxtimes$ , où  $v'$  représente le mot  $v$  auquel on a ajouté une tête de calcul. Le mot  $u$  contient donc l'état  $\blacksquare$  en son centre. Supposons que  $\mathcal{M}$  ne s'arrête pas sur  $v$ , alors pour toute configuration de  $[u]$ , le calcul entre l'état  $\blacksquare$  d'abscisse 0 et le premier état  $\boxtimes$  à sa droite ne s'arrêtera jamais. Soit parce que le calcul s'effectue, soit parce que la tête attend de disposer de plus d'espace. En particulier, pour tout site  $\langle 0, t \rangle, t \in \mathbb{N}$ , et toute configuration de  $[u]$ ,  $\mathcal{A}^t(c)_0 = \blacksquare$ . Donc  $\{\langle 0, t \rangle, t \in \mathbb{N}\} \subseteq \mathfrak{C}(u)$ .

Si en revanche,  $\mathcal{M}$  s'arrête sur  $v$  mais nécessite plus d'espace pour s'arrêter que  $|v|$ , on regarde deux configurations :  $c_0 = \mathbb{N}\square u\square\mathbb{N}$  et  $c_0 = \mathbb{N}\blacksquare u\blacksquare\mathbb{N}$ . Le diagramme espace-temps de  $\mathcal{A}$  sur la configuration initiale  $c_0$  est une simulation de  $\mathcal{M}$  sur l'entrée  $v$ . A chaque fois que la tête de calcul nécessitera de l'espace, elle pourra pousser l'état  $\boxtimes$  à sa droite, puisqu'il sera suivi par un  $\square$ . Donc la simulation ne sera pas interrompue et le calcul s'arrêtera de lui même. L'état  $\blacksquare$  en abscisse 0 sera détruit et remplacé par  $\square$ .

Avec  $c_1$  pour configuration initiale, en revanche, le calcul n'a pas assez d'espace pour s'effectuer entièrement, et l'état  $\boxtimes$  à droite ne peut pas être déplacé puisqu'il est suivi par  $\blacksquare$ . L'AC est alors dans une configuration stable, il n'évolue plus. Et en particulier, pour tout  $t \in \mathbb{N}$ ,  $\mathcal{A}^t(c_1)_0 = \blacksquare$ .

Supposons que l'on puisse, en se donnant un mot  $u \in X^*$ , décider si  $\{\langle 0, t \rangle, t \in \mathbb{N}\} \subseteq \mathfrak{C}(u, \mathcal{A})$ . Il s'ensuit que l'on peut décider si la machine codée dans l'automate s'arrête sur l'entrée codée par  $u$ . Ce qui contredit l'indécidabilité du problème de l'arrêt d'une machine de Turing.  $\square$

### 2.1.6 Etats dans un ensemble de conséquences

En regardant les conséquences d'un mot  $u$ , on va s'intéresser aux configurations contenant plusieurs occurrences de  $u$ . En particulier, les sites dont l'état est forcé par chacune des occurrences de  $u$  donnent des contraintes sur plusieurs sites des conséquences. On va aussi généraliser ce procédé en s'intéressant à des configurations périodiques dont la période contient  $u$ .

**Proposition 2.1.5.** *Pour un AC  $\mathcal{A}$  et un mot  $v$  sur son alphabet, il existe une prépériode  $p$  et une période  $\tau$  telles que :*

- $\forall x, x' \in \mathbb{Z}, x - x' \equiv 0[|v|] \Rightarrow \forall t \in \mathbb{N}, \mathcal{A}^t(v^{\mathbb{Z}})_x = \mathcal{A}^t(v^{\mathbb{Z}})_{x'}$
- $\forall t, t' \geq p, t - t' \equiv 0[\tau] \Rightarrow \forall x \in \mathbb{Z}, \mathcal{A}^t(v^{\mathbb{Z}})_x = \mathcal{A}^{t'}(v^{\mathbb{Z}})_x$

*Démonstration.* Comme  $v^{\mathbb{Z}}$  est une configuration périodique de période  $v$ , ses images successives par  $\mathcal{A}$  sont périodiques de périodes  $(v_t)_{t \in \mathbb{N}}$  avec  $|v_t| = |v|$  pour tout  $t$ . Ainsi  $|v|$  est une période spatiale pour  $\mathcal{A}^t(v^{\mathbb{Z}})$ .

Or  $X^{|v|}$  est fini, donc il existe  $t_1, t_2 \in \mathbb{N}$  tels que  $v_{t_1} = v_{t_2}$ , c'est à dire  $\mathcal{A}^{t_1}(v^{\mathbb{Z}}) = \mathcal{A}^{t_2}(v^{\mathbb{Z}})$ . On note donc  $\tau = t_2 - t_1$  et  $p = t_1$ . Comme  $\mathcal{A}^{t_1}(v^{\mathbb{Z}}) = \mathcal{A}^{t_2}(v^{\mathbb{Z}})$ , on note  $c = \mathcal{A}^p(v^{\mathbb{Z}})$  et on a aussi  $\mathcal{A}^\tau(c) = c$ .  $\square$

On définit maintenant les fonctions représentant les bords des conséquences, c'est à dire donnant les suites des sites extrêmes des conséquences d'un mot :

**Définition 2.1.2.** Pour  $\mathcal{A}$  un AC et  $u \in X^*$  un mot fini sur son alphabet, le bord gauche (resp. droit) des conséquences de  $u$  pour  $\mathcal{A}$  est :

$$\forall t \in \mathbb{N}, f_u^g(t) = \min_{\langle x, t \rangle \in \mathfrak{C}(u)} x \text{ (resp. } \forall t \in \mathbb{N}, f_u^d(t) = \max_{\langle x, t \rangle \in \mathfrak{C}(u)} x)$$

En s'intéressant maintenant aux états qui apparaissent dans l'ensemble des conséquences d'un mot, on obtient les propriétés suivantes :

**Proposition 2.1.6.** *Pour tout mot  $u \in X^l$ ,  $l \in \mathbb{N}$ , et pour  $t \in \mathbb{N}$ , si  $f_u^d(t) - f_u^g(t) \geq 2l$ , alors  $\exists s \in X, \forall \langle x, t \rangle \in \mathfrak{C}(u), \forall c \in [u], \mathcal{A}^t(c)_x = s$ .*

*Démonstration.* L'idée, illustrée par la figure 2.7, est de regarder une configuration avec deux occurrences de  $u$ . Un état en particulier est dans les conséquences de chacune de ces occurrences.

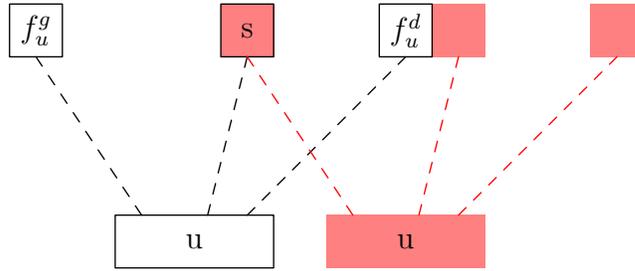


FIGURE 2.7 – Deux occurrences de  $u$  forcent l'état du site  $s$ .

Soient  $u \in X^l$  et  $t \in \mathbb{N}$  tels que  $f_u^d(t) - f_u^g(t) \geq 2l$ . Soit  $x \in \mathbb{Z}$  tel que  $\langle x, t \rangle \in \mathfrak{C}(u)$ . Supposons que  $f_u^d(t) - x \geq l$  (sinon  $|x - f_u^g(t)| \geq l$ ). On prend  $a \in X$  et on regarde la configuration  $c = {}^{\mathbb{Z}} a u a^{f_u^d(t)-x} u a^{\mathbb{Z}}$ . On va montrer que  $\mathcal{A}^t(c)_x = \mathcal{A}^t(c)_{f_u^d(t)}$ . Comme  $\langle f_u^d(t), t \rangle \in \mathfrak{C}(u)$ , il existe  $s \in X$  tel que  $\mathcal{A}^t(c)_{f_u^d(t)} = s$ . Comme  $\langle x, t \rangle \in \mathfrak{C}(u)$ , il existe  $s' \in X$  tel que  $\mathcal{A}^t(c)_x = s'$  et  $\mathcal{A}^t(c)_{x+(f_u^d(t)-x)} = s'$  (un site est forcé pour chaque occurrence de  $u$ ). Donc  $s = s'$  et  $\mathcal{A}^t(c)_x = \mathcal{A}^t(c)_{f_u^d(t)}$ . Par définition des conséquences, c'est vrai pour toute configuration.  $\square$

**Proposition 2.1.7.** *Pour tout AC  $\mathcal{A}$  et tout mot  $u \in X^*$ , il existe  $p, \tau \in \mathbb{N}$ , tel que  $\forall \langle x, t \rangle, \langle x', t' \rangle \in \mathfrak{C}(u)$  avec  $t, t' \geq p$ , on a :*

$$\forall c \in [u], (x - x') \equiv 0[|u|] \text{ et } (t - t') \equiv 0[\tau] \Rightarrow \mathcal{A}^t(c)_x = \mathcal{A}^{t'}(c)_{x'}$$

*Démonstration.* Par définition,  $\forall c, c' \in [u], \mathcal{A}^t(c)_x = \mathcal{A}^t(c')_x$  donc en particulier,  $\forall c \in [u], \mathcal{A}^t(c)_x = \mathcal{A}^t(u^{\mathbb{Z}})_x$ . il suffit donc de prouver la propriété pour  $u^{\mathbb{Z}}$ . Comme c'est une configuration périodique de période  $|u|$ , la proposition 2.1.5 permet de conclure.  $\square$

### 2.1.7 Ensembles finis

On traite le cas des ensembles finis séparément. On va en effet voir qu'ils sont tous réalisables modulo l'ajout des sites dont les antécédents sont tous dans l'ensemble.

**Définition 2.1.3.** Pour un ensemble fini  $C \subset \mathbb{Z} \times \mathbb{N}$ , on note  $l(C)$  sa largeur définie par  $l(C) = \max\{i \in \mathbb{Z}, \langle i, j \rangle \in C\} - \min\{i \in \mathbb{Z}, \langle i, j \rangle \in C\}$ .

**Proposition 2.1.8.** Soit  $C \subset \mathbb{Z} \times \mathbb{N}$  fini tel que si  $C$  contient tous les antécédents d'un site, il contient aussi ce site. On note  $m = \min_{\langle x, t \rangle \in C} (x)$  et  $M = \max_{\langle x, t \rangle \in C} (x)$ , alors il existe un AC  $\mathcal{A}$  de rayon 1 et un mot  $u$  sur son alphabet  $X$ , tels que :

- $|u| \leq l(C)$  ;
- $\mathfrak{C}(u, \mathcal{A}) = \{\langle x - \lfloor \frac{M+m}{2} \rfloor, t \rangle, \langle x, t \rangle \in C\} \cup E$   
où  $E = \left\{ \langle x, t \rangle, -\lfloor \frac{|u|-1}{2} \rfloor + t \leq x \leq \lfloor \frac{|u|}{2} \rfloor - t, x, t \in \mathbb{Z} \times \mathbb{N}^* \right\}$ .

*Démonstration.* On note  $B = \{\langle x - \lfloor \frac{M+m}{2} \rfloor, t \rangle, \langle x, t \rangle \in C\} \cup E$ .  $B$  est donc la réunion des sites de  $C$  décalés pour centrer  $C$  sur l'origine et des états de  $E$  qui composent le cône nécessairement inclus dans les conséquences de  $u$ . Soit  $t_0 = \max_{\langle x, t \rangle \in C} \{t\}$ . On définit l'ensemble  $X$  d'états comme suit :

- à tout site  $\langle x, t \rangle \in B$ , on associe un état  $a_{x,t}$ ,
- à tout site  $\langle x, t \rangle \notin B$  avec  $m \leq x \leq M$  et  $0 \leq t \leq t_0$  on associe deux états  $b_{x,t}$  et  $c_{x,t}$ ,
- on ajoute les états  $b_{x,0}$  pour  $m \leq x \leq M$ ,
- on ajoute deux états  $d_b$  et  $d_c$ .

On considère le mot  $u = a_{m,0}a_{m-1,0} \dots a_{M,0}$ . L'idée est qu'à partir du mot  $u$ , on connaît les coordonnées de chaque site de  $B$ . Dans le rectangle  $[m, M] \times [0, t_0]$ , les états contiendront toujours les coordonnées du site. Ensuite si un site doit être fixé, il sera dans l'état  $a_{x,t}$ , et sinon soit  $b_{x,t}$  soit  $c_{x,t}$ , ainsi, deux états différents seront possible et le site ne sera pas dans les conséquences.

On a un AC de rayon 1, donc chaque site a trois antécédents. La règle de transition ne regardera le plus souvent que l'état de la même cellule au temps précédent.  $d_b$  et  $d_c$  sont conservés, et les états  $e_{x,t}$  deviennent  $e_{x,t+1}$  où  $e$  peut être  $a, b$  ou  $c$ . Dans ce deuxième cas :

1. si  $\langle x, t+1 \rangle \in B$ , le site prend l'état  $a_{x,t+1}$ ,
2. si  $\langle x, t+1 \rangle \notin B$ , le site prend l'état  $b_{x,t+1}$  ou  $c_{x,t+1}$ , selon qu'il y a plus de  $b$  ou de  $c$  dans les antécédents ( $d_b$  et  $d_c$  comptent alors respectivement pour un  $b$  et un  $c$ ),
3. si  $t+1 > t_0$ , le site prend l'état  $d_b$  ou  $d_c$ , selon qu'il y a plus de  $b$  ou de  $c$  dans les antécédents.

Pour la règle 2, on accorde par exemple la priorité à  $b$  en cas de litige.

Ainsi, à partir d'une configuration contenant  $u$ , tous les états qui doivent être fixés le sont puisqu'ils sont nécessairement dans l'état  $a_{x,t}$ . Pour les autres, on considère les configurations  ${}^\omega d_b u d_b^\omega$  et  ${}^\omega d_c u d_c^\omega$ . Dans la première, il n'y aura ni état  $d_c$ , ni état  $c_{x,t}$ . Et inversement dans la seconde. De plus, les états  $a_{x,t}$  n'apparaissent que dans les sites de  $B$ , puisqu'ils ne peuvent être issus que d'états  $e_{x,t}$ . □

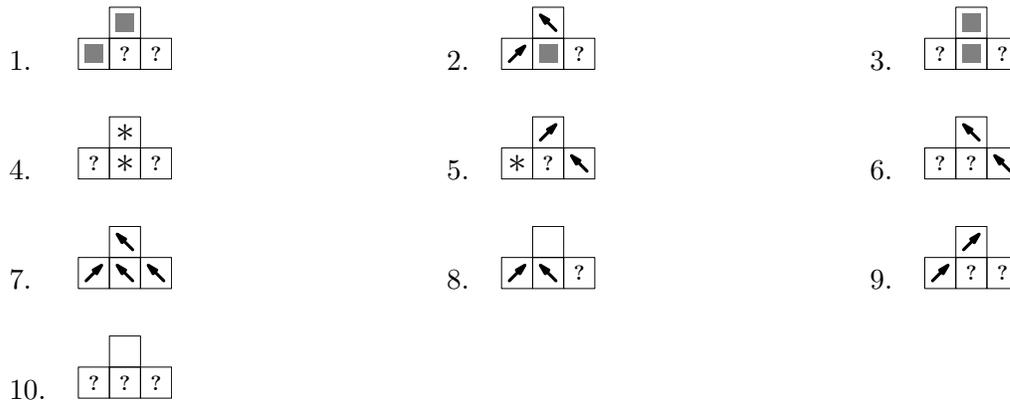
## 2.2 Conséquences paraboliques

On a montré qu'il est possible d'obtenir des ensembles de conséquences prenant beaucoup de formes simples différentes, on va chercher ici à construire des formes plus sophistiquées, en particulier un ensemble limité par une parabole :

**Proposition 2.2.1.** *Il existe un  $AC$  et un mot  $u$  sur son alphabet tels que*

$$\mathfrak{C}(u, \mathcal{A}) = \{ \langle x, t \rangle, x \leq t < (x + 1)^2 \}.$$

*Démonstration.* La figure 2.8 illustre la construction que l'on va décrire. On va construire un automate de rayon 1 à cinq états :  $X = \{\blacksquare, \square, *, \nearrow, \searrow\}$ . On définit la règle locale avec les contraintes suivantes dans l'ordre décroissant de priorité :



L'idée est d'avoir un état  $\blacksquare$  qui se propage toujours vers la droite et vers le haut lorsqu'il ne reçoit pas de signal  $\nearrow$  sur sa gauche. On fait rebondir  $\nearrow$  et  $\searrow$  sur les états  $\blacksquare$  à droite et  $*$  (qui reste en place) à gauche. Le signal rebondit donc entre la limite de la zone  $\blacksquare$  et un point fixe. C'est une construction classique d'automate cellulaire

pour tracer une parabole. Fischer [11] utilise par exemple une construction similaire pour marquer les entiers premiers.

On note  $E = \{\langle x, t \rangle, 0 \leq x \leq t < (x+1)^2\}$  et on choisit  $u = \blacksquare$ . On commence par montrer que  $\mathfrak{C}(u) \subseteq E$ . Pour cela, on regarde les sites qui sont dans le même état pour les configurations  $\blacksquare^{\mathbb{Z}}$  et  $\mathbb{N} \square \begin{smallmatrix} * \\ \nearrow \end{smallmatrix} \blacksquare \mathbb{N}$ . Les images successives de  $\blacksquare^{\mathbb{Z}}$  sont toutes  $\blacksquare^{\mathbb{Z}}$ . Les conséquences de  $u$  sont donc parmi les sites dans l'état  $\blacksquare$  dans le diagramme espace-temps associé à la configuration initiale  $\mathbb{N} \square \begin{smallmatrix} * \\ \nearrow \end{smallmatrix} \blacksquare \mathbb{N}$ . Sur cette configuration, rien ne vient perturber le comportement illustré par la figure 2.8, un signal formé par les états  $\nearrow$  et  $\searrow$  rebondit entre  $*$  et la frontière de la zone remplie par  $\blacksquare$ . A chaque fois qu'elle atteint cette frontière, elle détruit un état  $\blacksquare$  avant de repartir. Ainsi, le signal parcourt une distance plus grande à chaque aller-retour. A l'abscisse  $x \in \mathbb{N}$ , un état  $\blacksquare$  est produit à l'étape  $x$ , et il est détruit à l'étape  $(x+1)^2$  donc  $\mathfrak{C}(u) \subseteq E$ .

Pour montrer l'inclusion  $E \subseteq \mathfrak{C}(u)$ , on considère une configuration  $c \in [u]$ , et on montre que  $\forall \langle x, t \rangle \in E, \mathcal{A}^t(c)_x = \blacksquare$ .

POINT 1: Soit  $k > 0$  minimal tel que  $\mathcal{A}^k(c)_0 = \blacksquare$  et  $\forall q \in \mathbb{N}, 0 < q < k, \mathcal{A}^q(c)_0 \neq \blacksquare$ . On montre que  $\forall \langle x, t \rangle \in E, \text{ si } t \leq x+k \text{ alors } \mathcal{A}^t(c)_x = \blacksquare$ .

*Preuve:* Lorsqu'un site est dans l'état  $\blacksquare$ , le site situé au dessus à droite dans le diagramme espace-temps l'est aussi, chaque état  $\blacksquare$  implique donc que la diagonale vers la droite soit dans le même état (si  $\mathcal{A}^t(c)_x = \blacksquare$ , alors  $\forall y, \mathcal{A}^{t+y}(c)_{x+y} = \blacksquare$ ). De plus,  $\blacksquare$  est produit par la règle locale seulement si l'antécédent gauche ou central était dans l'état  $\blacksquare$ . Donc grâce à la définition de  $k$ , on sait que  $c_{-k} = \blacksquare$  et  $\forall 0 < q < k, c_{-q} \neq \blacksquare$ .

Supposons  $k > 1$  (si  $k = 1$ , il n'y a rien à montrer). Comme  $\mathcal{A}(c)_0 \neq \blacksquare$ , nécessairement  $c_{-1} = \nearrow$ . Donc  $\mathcal{A}(c)_0 = \searrow$ . Et l'état  $\searrow$  avance vers la gauche en éliminant les états  $\nearrow$ , jusqu'à rencontrer un état  $\blacksquare$  ou  $*$ . Si c'est un état  $\blacksquare$ , alors  $\searrow$  disparaît et il n'y a que des états  $\square$  sur la gauche jusqu'au  $\blacksquare$ . En particulier,  $\forall 1 \leq q \leq k+1, \mathcal{A}^q(c)_1 = \blacksquare$ , et nécessairement tous les sites sur les diagonales partant de ces sites sont aussi dans l'état  $\blacksquare$ .

Si  $\searrow$  rencontre un état  $*$ , il se transforme en  $\nearrow$  puis retourne vers la droite. Il y aura donc un va et vient (avec les états  $\nearrow$  et  $\searrow$ ) entre l'état  $*$  et la frontière de la zone dans l'état  $\blacksquare$ . A chaque fois que  $\nearrow$  rencontre un état  $\blacksquare$ , il le supprime, donc la distance à parcourir augmente d'une unité à chaque parcours. Comme  $*$  reste inchangé tant qu'il n'y a pas de  $\blacksquare$  à gauche, il n'est plus nécessaire de regarder ce qui se passe à gauche.

Plus  $*$  est proche de la frontière, plus les  $\blacksquare$  seront supprimés vite, donc, on s'intéresse au cas où  $c_{-2} = *$ . On a alors le diagramme illustré par la figure 2.9.

Dans ce cas, on regarde la suite  $(s_n)_{n \in \mathbb{N}}$  telle que les sites  $\langle n, s_n \rangle$  soient les sites où un  $\blacksquare$  est supprimé. Entre  $s_{n-1}$  et  $s_n$ , il faut que la distance soit parcourue deux fois entre la frontière  $\blacksquare$  et  $*$  à l'abscisse  $-2$ . Cela prend  $2n+1$  étapes (le signal ne

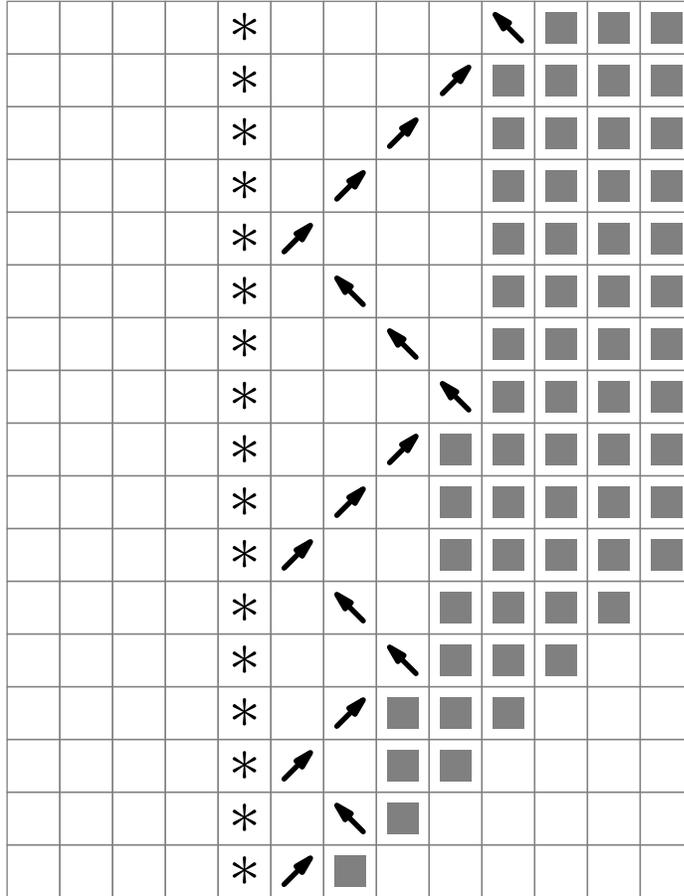


FIGURE 2.8 –

passe qu'une étape à l'abscisse  $-1$ ). Donc  $s_n = s_{n-1} + 2n + 1$ . Comme  $s_0 = 1$ , on a exactement  $s_n = (n + 1)^2$ . Comme la diagonale  $\{\langle x, t \rangle, t = x + k\}$  est dans l'état  $\blacksquare$ , les états à gauche de cette diagonale n'influencent pas le comportement à droite. Donc  $\forall \langle x, t \rangle \in E$  avec  $t \leq x + k$ , on a  $\mathcal{A}^t(c)_x = \blacksquare$ .

◇ Point 1

On peut maintenant réutiliser ce procédé avec le site  $\langle 0, k \rangle$  puis itérer en considérant à chaque fois le premier état  $\blacksquare$  vu à l'abscisse 0 après un autre état.

On définit la suite  $(k_n)_n$  en prenant  $k_0 = 0$ ,  $k_1 = k$ , puis  $k_{n+1} > k_n$  minimal tel que  $\mathcal{A}^{k_{n+1}}(c)_0 = \blacksquare$  et  $\mathcal{A}^{k_{n+1}-1}(c)_0 \neq \blacksquare$ .

POINT 2:  $\forall n \in \mathbb{N}, \forall \langle x, t \rangle \in E$ , si  $t \leq x + k_n$  alors  $\mathcal{A}^t(c)_x = \blacksquare$

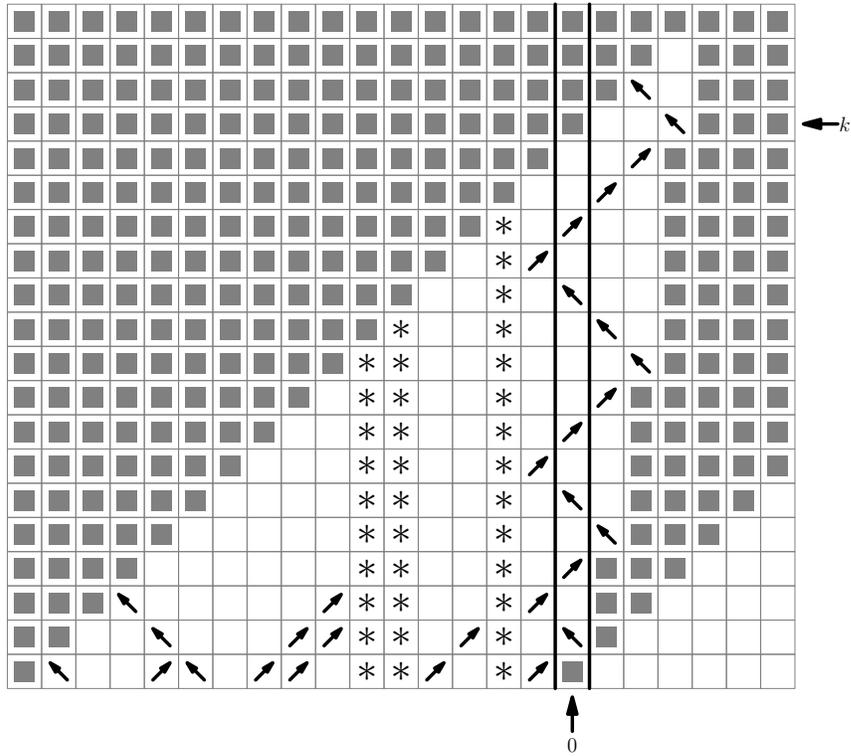


FIGURE 2.9 –

*Preuve:* Supposons la propriété vraie pour tout  $i \leq n$ . Grâce au point 1, on a :

$$\forall (x, t), x + k_n \leq t \leq x + k_{n+1}, t \leq x^2 + k_n \Rightarrow \mathcal{A}^t(c)_x = \blacksquare$$

En particulier,  $\forall (x, t) \in E$ , si  $x + k_n \leq t \leq x + k_{n+1}$  alors  $\mathcal{A}^t(c)_x = \blacksquare$ .

◇ Point 2

Pour clore la preuve de la proposition, il reste à considérer le cas où la suite  $(k_n)_n$  est finie, c'est à dire qu'à partir d'une certaine étape, on ne voit plus d'état  $\blacksquare$  sur la colonne d'abscisse 0. Dans ce cas, le fonctionnement vu dans la preuve du point 1 se perpétue indéfiniment et le résultat est vrai sans borne sur  $x$ . □

On peut donc obtenir un ensemble de conséquences limité par une parabole. La méthode utilisée peut se généraliser à d'autres fonctions. le principe général consiste à dessiner une grande zone de sites ayant l'état  $\blacksquare$  a priori, puis de creuser cette zone en envoyant des signaux par la gauche. La condition à remplir est de réussir à générer

au maximum le nombre de signaux dessinant exactement notre fonction : pour que la limite soit la représentation de la fonction  $f$ , il faut que le  $x^e$  signal arrive au temps  $f(x)$ . La difficulté est qu'il ne doit pas exister de configuration générant plus de signaux.

On peut aussi s'intéresser à obtenir un ensemble de conséquences dont la frontière droite est une parabole. Une construction simple pour tracer la même parabole mais par la droite utilise un procédé similaire : un signal alterne entre la parabole et une référence en effectuant un trajet plus long d'une unité à chaque fois. Dans ce cas, la parabole avance vers la référence au lieu de s'en éloigner, donc pour augmenter l'écart entre les deux, il faut déplacer la référence de deux cellules vers la gauche à chaque rebond du signal. Cette référence forme donc elle-même une parabole. Cette méthode présente un inconvénient : en l'absence de signal, on trace la ligne verticale. Supposons maintenant que l'on essaie de remplir l'ensemble de conséquences d'états  $\blacksquare$ . Tant que le calcul s'effectue, la limite de la zone  $\blacksquare$  est la parabole désirée, mais si le calcul est interrompu, cette limite devient la droite verticale. Les sites de l'ensemble des conséquences ne sont donc plus tous dans l'état  $\blacksquare$ . Pour contourner cette difficulté, on va assurer la quasi-intégrité de la construction.

## 2.3 Compteurs

On a vu dans la section précédente qu'il faut parfois pouvoir assurer que le calcul est effectué quelle que soit la configuration initiale. On montre ici qu'il est possible de protéger une aire de calcul en empêchant tout signal indésirable de rentrer. Une idée naturelle est de protéger un mot entre deux signaux allant à vitesse maximale vers l'extérieur et effaçant tout ce qu'ils rencontrent. Entre ces signaux, le calcul peut s'effectuer sans être perturbé. Seulement, ces signaux peuvent exister dans la configuration initiale en d'autres endroits. En particulier des signaux identiques peuvent rencontrer ceux qui protègent le calcul. Il est impossible d'accorder la priorité à l'un ou à l'autre, et on est ainsi obligé, soit de supprimer la protection, soit de fixer une barrière qui ne bougera plus. Ces deux cas sont problématiques puisque le calcul peut nécessiter un espace non borné.

### 2.3.1 Principe

On cherche ici à ce qu'un mot produise un espace de calcul protégé. On a vu qu'il n'est pas suffisant d'encadrer cet espace par de simples signaux. Il est notamment impossible d'éviter le problème posé par la répétition de ce mot sur la configuration initiale. Pour s'autoriser des grands calculs, on est de plus obligés d'utiliser un espace de calcul non borné. Il est donc nécessaire de « fusionner » les espaces de calcul créés par deux occurrences du mot initial. On verra comment gérer ce double calcul, en

utilisant le fait que ce sont deux évolutions contrôlées et semblables, l'une translatée par rapport à l'autre.

Il reste à assurer que seule une copie du mot initial crée un espace de calcul capable d'interférer avec le calcul d'origine. Pour cela, on utilise un état d'Eden, c'est à dire un état  $\boxed{*}$  qui n'est produit par aucune instance de la règle locale. Cet état sera le mot générant des conséquences. Il génère des protections autour de l'espace central qui servira au calcul. Ces protections doivent disparaître lorsqu'elles rencontrent des copies d'elles mêmes provenant de la même source, c'est à dire de l'état  $\boxed{*}$  sur la configuration initiale. En revanche, il faut pouvoir différencier des protections déjà présentes sur la configuration initiale et des protections créées par  $\boxed{*}$ . On utilise alors ce qui les particularise : les protections créées par  $\boxed{*}$  sont les plus jeunes. Il suffit donc de compter leur âge depuis leur formation et de comparer en cas de rencontre.

### 2.3.2 Construction

L'état  $\boxed{*}$  va créer deux constructions  $C_g$  et  $C_d$  respectivement sur sa gauche et sur sa droite, qui contiendront chacune un compteur.  $C_g$  et  $C_d$  étant totalement symétriques, on se contentera de décrire  $C_d$ . Le compteur interne de  $C_d$  n'est pas borné, il nécessite donc un espace non borné. On va l'encadrer entre deux signaux  $S_{d1}$  et  $S_{d2}$  de vitesses  $v_1 > v_2 > 0$  (respectivement  $S_{g1}$  et  $S_{g2}$  avec les vitesses  $-v_1$  et  $-v_2$  à gauche). On compte à l'aide d'un compteur unaire d'arité  $K$ , les impulsions étant envoyées par le signal extérieur  $S_{d1}$ . Ainsi, au temps  $t$ , on dispose de l'espace  $[(v_2 - v_1)t]$  entre les deux signaux, on peut donc compter jusqu'à  $K[(v_2 - v_1)t]$ , c'est-à-dire connaître l'âge du compteur depuis sa création. On choisira  $K$  en fonction de  $v_1$  et  $v_2$ .

La règle ensuite, est que seul un signal extérieur ( $S_{d1}$  ou  $S_{g1}$ ) peut passer à travers un signal extérieur opposé. Dans ce cas, il faudra gérer la collision des compteurs, comparer leurs âges et décider de garder le plus jeune ou de supprimer les protections, s'ils ont le même âge. Dans tous les autres cas, un signal extérieur avance sans être perturbé.

### 2.3.3 Collisions

Il reste donc à s'assurer que l'on peut comparer l'âge de deux compteurs qui se rencontrent, et en détruire au moins un, avant que les espaces de calculs protégés ne soient perturbés. Il faudra aussi vérifier que cela ne perturbe pas le fonctionnement du compteur restant le cas échéant. Puisque les compteurs sont unaires, il suffit de comparer leurs longueurs et en cas d'égalité, leurs derniers symboles. On décrit maintenant le processus de comparaison illustré par la figure 2.11 :



le compteur gauche par exemple :  $C_d$  arrive avant  $C_g$ . On va donner les contraintes nécessaires sur les vitesses  $v_1$  et  $v_2$  (voir figure 2.11).

Pour le calcul, on place l'origine en  $A$ , et on considère des AC de rayon  $r$  (donc la vitesse maximale est  $r$  aussi et on note  $p = \frac{1}{r}$ ), on obtient alors les équations de droite et les points suivants :

$$(AB_d) : y = -px$$

$$(B_dE) : y = \frac{1}{v_2}x + \frac{1}{v_2}a \quad B_d = \left(-\frac{a}{pv_2+1}, \frac{pa}{pv_2+1}\right)$$

$$(B_dC_d) : y = px + 2\frac{pa}{pv_2+1} \quad C_d = \left(0, 2\frac{pa}{pv_2+1}\right)$$

$$(C_dD) : y = -px + 2\frac{pa}{pv_2+1}$$

$$(AE) : y = -\frac{1}{v_1}x \quad E = \left(-\frac{v_1a}{v_2+v_1}, \frac{a}{v_2+v_1}\right)$$

La comparaison réussit si la droite  $(C_dD)$  coupe la droite  $(B_dE)$  en  $F$  avec  $x_F \leq x_E$ . Or  $x_F = \frac{pv_2a-a}{(pv_2+1)^2}$ . Et donc on obtient la contrainte  $\frac{pv_2a-a}{(pv_2+1)^2} \leq -\frac{v_1a}{v_2+v_1}$ . C'est à dire  $v_1(pv_2 - 1 + (pv_2 + 1)^2) \leq -v_2(pv_2 - 1)$ . Or par définition, on veut  $v_1 < v_2$ , donc on obtient  $v_2 < \frac{r}{2+\sqrt{5}}$ .

On a vu qu'envoyer des signaux était aisé lorsque ceux-ci sont de vitesse rationnelle. Pour  $r = 1$  par exemple, on prend donc  $v_1 = \frac{1}{4}$  et  $v_2 = \frac{1}{5}$ , ainsi tous les signaux ont une vitesse rationnelle : 1, 1/4 ou 1/5. Il reste à donner l'arité du compteur unaire. On prend par exemple  $K = \frac{1}{v_2-v_1} = 20$ , ainsi, on dispose de la place suffisante entre les deux signaux pour compter l'âge de la structure.

Ce choix des vitesses  $v_1$  et  $v_2$  est arbitraire, tous les autres choix respectant ces contraintes fonctionneraient aussi, en utilisant une arité adaptée. De même le rayon de l'AC et donc la vitesse maximale peuvent être différents, en particulier, avec un grand rayon, on pourra obtenir une vitesse  $v_2$  aussi faible que voulue et ainsi se libérer un grand espace de calcul entre les compteurs.

*Remarque 2.3.2.* Au cours des premières étapes de calcul de l'AC, les compteurs n'ont pas encore acquis la structure permettant d'effectuer les comparaisons. En particulier, les signaux intérieur et extérieur des compteurs ne se sont pas forcément encore séparés, ou les signaux intérieurs des deux compteurs sont encore superposés. Il faudra donc gérer des cas particuliers, qui ne se produisent que dans les premières étapes de calcul, on aura donc recours à de nombreux états transitoires qui n'apparaîtront plus par la suite.

### 2.3.4 Utilisation

On va maintenant utiliser l'espace de calcul protégé par les compteurs. En particulier, on va construire un AC  $\mathcal{A}$  et un mot  $u$  sur son alphabet ayant comme ensemble de conséquences  $\{\langle x, t \rangle, x \geq 0, t \geq x^2\}$ . On va utiliser la construction envisagée précédemment : un signal rebondit entre la limite de la zone  $\blacksquare$  qui marque l'ensemble des conséquences, et une référence qui trace elle même une parabole. Pour cela, la référence est déplacée de deux cellules vers la droite lorsque le signal l'atteint et la limite de la zone  $\blacksquare$  d'une seule cellule. La figure 2.12 illustre cette construction.

On a vu que cette construction seule ne fonctionnait pas, car il faut assurer que le calcul est effectué. En particulier, un mur  $\blacksquare$  peut l'interrompre, et rien ne garantit alors que l'ensemble  $\{\langle x, t \rangle, x \geq 0, t \geq x^2\}$  ne contienne que l'état  $\blacksquare$ . Pour remédier à ce problème, on va insérer cette construction entre des compteurs. Il faut utiliser des compteurs suffisamment larges (avec des pentes suffisamment faibles) pour que tous les sites utiles à construire la parabole soient à l'intérieur des compteurs. C'est possible en prenant un AC avec un grand rayon. L'état  $\boxtimes$  lance alors la construction interne en même temps que les compteurs.

La construction intérieure n'est donc pas perturbée jusqu'à ce qu'un des compteurs qui la protègent rencontre un autre compteur créé par un signal  $\boxtimes$  à l'étape initiale. Ces deux compteurs vont disparaître, et on aura alors une nouvelle zone protégée par des compteurs, mais contenant cette fois deux constructions identiques de parabole comme vu en figure 2.13. Puisqu'elles sont identiques, la parabole de gauche ne dépassera jamais celle de droite, il y aura donc une zone  $\blacksquare$  étendue contenant  $\{\langle x, t \rangle, x \geq 0, t \geq x^2\}$ . Finalement, soit la construction n'est pas perturbée, et la parabole est dessinée comme sur la figure 2.12 entre des compteurs, soit la construction est perturbée, auquel cas, tous les sites de l'ensemble de conséquences sont dans l'état  $\blacksquare$ , mais le dessin de la parabole n'est plus nécessairement apparent.

On obtient encore une fois un ensemble de conséquences de largeur non bornée. Pour autant, la largeur de l'ensemble de conséquences n'est pas nécessairement aussi grande. Pour ne garder que la parabole dans l'ensemble de conséquences par exemple, on peut utiliser les deux constructions décrites donnant des ensembles de conséquences dont les limites gauche et droite sont cette même parabole. On utilise la proposition 2.1.2 pour obtenir l'intersection des deux ensembles de conséquences. On obtient ainsi un ensemble de conséquences qui ne contient qu'une parabole.

### 2.3.5 Généralisation

Cela ne fonctionne que parce que les compteurs protègent de tout ce qui n'est pas rigoureusement identique. On peut utiliser cette technique pour construire beaucoup de fonctions monotones comme bordure d'ensembles de conséquences. Il suffit en effet

de pouvoir construire une fonction, en plaçant tout le calcul d'un seul côté. De l'autre côté, on remplit avec des états  $\blacksquare$  jusqu'à une droite verticale ou une diagonale par exemple. On peut ainsi construire des ensembles de conséquences limités par presque toutes les fonctions constructibles en signal comme défini par J. Mazoyer et V. Terrier [26].

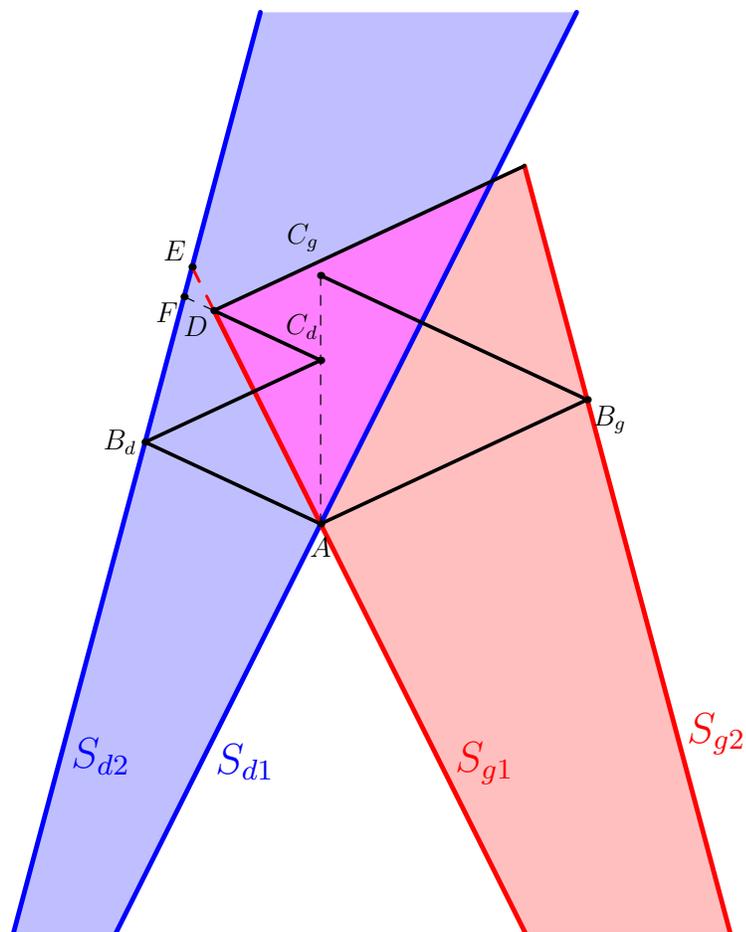


FIGURE 2.11 – Collision d'un compteur gauche et d'un compteur droit plus âgé.

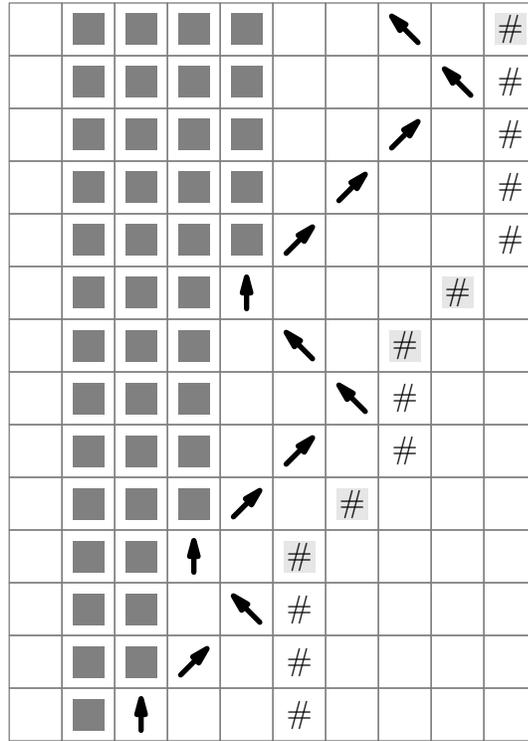


FIGURE 2.12 -

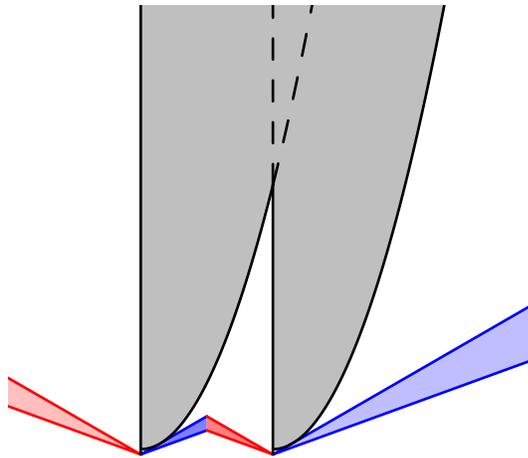


FIGURE 2.13 -

## Chapitre 3

# Directions bloquantes

Dans la partie précédente, on a vu comment il était possible de forcer les états de certains sites du diagramme espace-temps avec un nombre fini de contraintes sur la configuration initiale. L'information détenue par ces contraintes permettait de négliger tous les autres flux d'informations qui auraient pu atteindre les sites déterminés. On va utiliser cette propriété pour étudier la dynamique de certains de ces automates cellulaires. Lorsque, comme dans les exemples vus dans la partie précédente, les conséquences d'un mot forment un espace suffisamment large et continu, elles séparent le diagramme espace-temps en deux composantes entre lesquelles il n'y a pas d'interactions.

Dans ce cas, les flux d'informations sont interrompus par l'ensemble de conséquences qu'on appelle un *mur*, et qui correspond dans certains cas à la notion de *point d'équicontinuité*. Par conséquent, seule une direction parallèle au mur permet de transmettre des informations sans qu'elles risquent d'être bloquées. Étudier l'ensemble des directions selon lesquelles il existe un mur donne donc en complément des informations sur la manière de transmettre l'information.

La dynamique des automates cellulaires a été étudiée par P. Kůrka [22], F. Blanchard et A. Maass notamment [2]. Puis M. Sablik a généralisé cette notion à des directions linéaires [37]. On l'étend ici à des directions quelconques.

Les résultats de ce chapitre ont été publiés dans [10].

### 3.1 Définitions

On s'intéresse aux directions selon lesquelles on peut interrompre le transfert d'informations. On ne fait aucune restriction a priori sur les fonctions concernées : une direction est donc une fonction  $h : \mathbb{N} \mapsto \mathbb{Z}$  et on note  $\mathcal{F} = \{h : \mathbb{N} \mapsto \mathbb{Z}\}$ .

On dira qu'un mot est bloquant selon la direction  $h$  si dans toute configuration

initiale contenant ce mot, aucun changement à droite de ce mot n'a de répercussion dans le diagramme espace-temps à gauche de la courbe de  $h$  et inversement.

**Définition 3.1.1.** Un *mot bloquant* de direction  $h \in \mathcal{F}$  pour un AC  $\mathcal{A}$  est un mot  $u \in X^*$  tel que pour toutes configurations  $c, c' \in [u]$  :

- $\forall x < 0, c_x = c'_x \Rightarrow \forall (x, t) \in \mathbb{Z} \times \mathbb{N}$  tel que  $x \leq h(t) \Rightarrow \mathcal{A}^t(c)_x = \mathcal{A}^t(c')_x$
- $\forall x \geq 0, c_x = c'_x \Rightarrow \forall (x, t) \in \mathbb{Z} \times \mathbb{N}$  tel que  $x > h(t) \Rightarrow \mathcal{A}^t(c)_x = \mathcal{A}^t(c')_x$

On va noter  $Bl^h(\mathcal{A}) \subseteq X^{\mathbb{Z}}$  l'ensemble des configurations au centre desquelles se trouve un mot bloquant selon la direction  $h$  :  $Bl^h(\mathcal{A}) = \{c \in X^{\mathbb{Z}}, \exists l \in \mathbb{N}, c_{[-l, l]}$  bloquant selon  $h$  }.

Un mot bloquant de direction  $h$  génère un *mur* de direction  $h$ , qui sera porté par la courbe de  $h$ . Le même mot bloquant placé à l'abscisse  $x_0$  générera un mot bloquant de direction  $(h+x_0)$  et un mot contenant un mot bloquant est lui même bloquant. Un mur coupe donc bien tout transfert d'information entre les deux espaces qu'il sépare. Il existe aussi un point de vue plus constructif. En effet, si les conséquences d'un mot forment une bande de largeur suffisante, ces conséquences ont le comportement d'un mur. On peut donc donner une définition de mur d'épaisseur  $e : \mathbb{N} \mapsto \mathbb{N}$ , dont on vérifiera ensuite qu'elle est cohérente avec la définition originelle.

**Définition 3.1.2.** On dit qu'un mot  $u$  génère une *cloison* de direction  $h \in \mathcal{F}$  et d'épaisseur  $e : \mathbb{N} \mapsto \mathbb{N}$  lorsqu'il existe deux fonctions  $h_g, h_d : \mathbb{Z} \mapsto \mathbb{N}$  telles que :

- $\forall t \in \mathbb{N}, h_g(t) \leq h(t) \leq h_d(t)$  ;
- $\forall t \in \mathbb{N}, e(t) = h_d(t) - h_g(t)$  ;
- $\{\langle x, t \rangle, t \in \mathbb{N}, h_g(t) \leq x \leq h_d(t)\} \subseteq \mathfrak{C}(u)$ .

Lorsque cette épaisseur est suffisante, elle coupe le transfert d'information :

**Définition 3.1.3.** On dit qu'un mot  $u$  génère un *mur* de direction  $h \in \mathcal{F}$  et d'épaisseur  $e : \mathbb{N} \mapsto \mathbb{N}$  lorsqu'il génère une cloison d'épaisseur  $e$  avec deux fonctions  $h_g, h_d : \mathbb{Z} \mapsto \mathbb{N}$  telles que :

- $\forall t \in \mathbb{N}, h_g(t) \leq h(t) \leq h_d(t)$  ;
- $\forall t \in \mathbb{N}, e(t) = h_d(t) - h_g(t)$  ;
- $\{\langle x, t \rangle, t \in \mathbb{N}, h_g(t) \leq x \leq h_d(t)\} \subseteq \mathfrak{C}(u)$  ;
- $\forall t \in \mathbb{N}, h_g(t+1) \leq h_d(t) + 1 - r$  ;
- $\forall t \in \mathbb{N}, h_d(t+1) \geq h_g(t) - 1 + r$  ;

où  $r$  est le rayon de l'AC.

La figure 3.1 illustre cette définition. En particulier, pour tout mot bloquant  $u$  d'épaisseur  $e : \mathbb{N} \mapsto \mathbb{N}$ ,  $\{\langle h(t), t \rangle, t \in \mathbb{N}\} \subseteq \mathfrak{C}(u)$ .

Lorsque  $e$  vérifie toutes ces propriétés, on dit que c'est une épaisseur *acceptable* pour le mot  $u$  et la direction  $h$ .

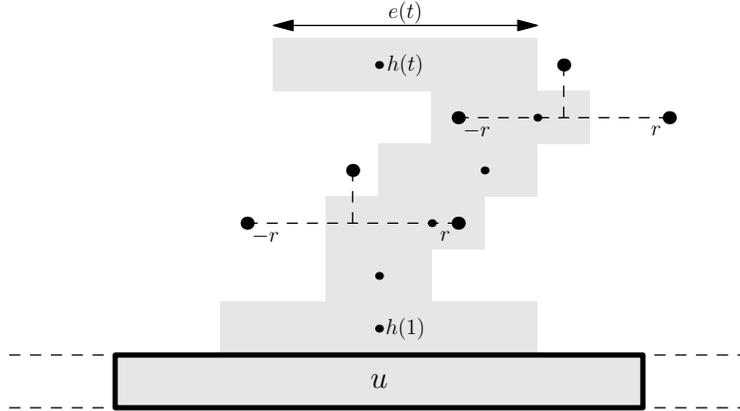


FIGURE 3.1 – En gris, les conséquences du mot  $u$  pendant les premières étapes, une cellule hors de la zone grise ne doit pas avoir d'antécédent de l'autre côté.

**Proposition 3.1.1.** *Un mot qui génère un mur de direction  $h \in \mathcal{F}$  et d'épaisseur  $e$  est un mot bloquant de direction  $h$ .*

*Démonstration.* Soit  $u \in X^{\mathbb{Z}}$  un mot qui génère un mur de direction  $h$  et d'épaisseur  $e$ , et  $c, c'$  deux configurations de  $[u]$ . Supposons que  $\forall x < 0, c_x = c'_x$ . On note  $h_g$  et  $h_d$  les fonctions définissant  $e$ . On va montrer la propriété  $\forall x \leq h_d(t), \mathcal{A}^t(c)_x = \mathcal{A}^t(c')_x$  par récurrence sur  $t$ .

Étant trivialement vraie sur la configuration initiale, on suppose qu'elle est vraie pour  $t \leq t_0$ . On prend  $x \leq h_d(t_0 + 1)$ . Alors soit  $x \geq h_g(t_0 + 1)$  auquel cas  $\langle x, t_0 + 1 \rangle \in \mathfrak{C}(u)$  et donc  $\mathcal{A}^{t_0+1}(c)_x = \mathcal{A}^{t_0+1}(c')_x$ , soit  $x < h_g(t_0 + 1)$ . Pour ce deuxième cas, on regarde les antécédents du site  $\langle x, t_0 + 1 \rangle$ , qui sont les sites  $\langle x + y, t_0 \rangle, -r \leq y \leq r$ . Or pour tout tel  $y, x + y < h_g(t_0 + 1) + r$ . Comme  $u$  génère un mur de direction  $h$  et d'épaisseur  $e$ , on sait que  $h_g(t_0 + 1) \leq h_d(t_0) + 1 - r$ , il s'ensuit que  $x + y \leq h_d(t_0)$ , et par hypothèse de récurrence  $\mathcal{A}^{t_0}(c)_{x+y} = \mathcal{A}^{t_0}(c')_{x+y}$ . Cela garantit que  $\mathcal{A}^{t_0+1}(c)_x = \mathcal{A}^{t_0+1}(c')_x$ . Un raisonnement symétrique donne le résultat pour  $c, c' \in [u]$  avec  $\forall x > 0, c_x = c'_x$ .  $\square$

Naturellement, pour un mot bloquant  $u$  et tous mots  $v$  et  $w$ , le mot  $vuw$  est lui aussi bloquant. Cependant, si  $u$  génère un mur d'épaisseur  $e$ , l'épaisseur de  $vuw$  n'est pas nécessairement plus que  $e$ . On va maintenant montrer que l'on peut construire un mot générant un mur d'épaisseur arbitraire à partir d'un mot bloquant.

**Lemme 3.1.2.** *Soit  $u \in X^*$  un mot bloquant de direction  $h \in \mathcal{F}$ , pour tout mot  $v \in X^*$  tel que  $|u| + |v| \geq 2r$ , le mot  $uvu$  génère une cloison de même direction  $h$  et d'épaisseur  $e$  telle que  $\forall t \in \mathbb{N}, e(t) \geq |u| + |v|$ .*

*Démonstration.* Pour une configuration de  $[uvu]$ , les positions des deux occurrences de  $u$  sont  $x_0 = -\left(\lfloor \frac{|v|}{2} \rfloor + \lceil \frac{|u|}{2} \rceil\right)$  et  $x_1 = \left(\lceil \frac{|v|}{2} \rceil + \lfloor \frac{|u|}{2} \rfloor\right)$ . On définit  $h_g$  et  $h_d$  par :

- $h_g(0) = x_0$  et  $h_d(0) = x_1$  ;
- $\forall t \in \mathbb{N}, h_g(t+1) = \min(h(t+1) + x_0, h_g(t) + r)$  ;
- $\forall t \in \mathbb{N}, h_d(t+1) = \max(h(t+1) + x_1, h_d(t) - r)$ .

Les fonctions  $h_g$  et  $h_d$  suivent donc les courbes de  $h$  décalées, mais elles ne dépassent pas la vitesse  $r$ . Lorsque la variation de  $h$  est supérieure à  $r$ ,  $h_g$  et  $h_d$  prennent donc du « retard », qu'elles pourront peut être rattrapper par la suite.

Clairement,  $\forall t \in \mathbb{N}, e(t) \geq x_1 - x_0$ . On peut vérifier que  $\forall t \in \mathbb{N}, h_g(t+1) \leq h_d(t) + 1 - r$ , car  $h_d(t) - h_g(t) \geq 2r$ . Et de même  $\forall t \in \mathbb{N}, h_d(t+1) \geq h_g(t) - 1 + r$ .

On va montrer par récurrence que

$$\forall t \in \mathbb{N}, \forall x, h_g(t) \leq x \leq h_d(t), \langle x, t \rangle \in \mathfrak{C}(uvu)$$

On suppose donc que la propriété est vérifiée pour  $t \leq t_0$ . On prend  $h(t_0+1) \leq x \leq h_d(t_0+1)$ , la preuve est symétrique pour  $h_g$ .

Soient  $c$  et  $c'$  des configurations dans  $[uvu]$ , on construit  $c'' \in [uvu]$  telle que  $\forall z > 0, c''_z = c_z$  et  $\forall z < 0, c''_z = c'_z$ . Donc  $c, c'$  et  $c''$  contiennent des mots bloquants de directions  $h + x_0$  et  $h + x_1$ . Si  $h_d(t_0+1) = h(t_0+1) + x_1$ , le résultat se prouve en disant que  $uvu$  est bloquant de directions  $h + x_0$  et  $h + x_1$ , donc

$$\mathcal{A}^{t_0+1}(c')_x = \mathcal{A}^{t_0+1}(c'')_x \text{ et } \mathcal{A}^{t_0+1}(c)_x = \mathcal{A}^{t_0+1}(c'')_x$$

Sinon, comme  $uvu$  est bloquant de direction  $h + x_0$  :

$$\mathcal{A}^{t_0+1}(c)_x = \mathcal{A}^{t_0+1}(c'')_x$$

On va maintenant montrer que les antécédents de  $\langle x, t_0+1 \rangle$  sont dans le même état dans les diagrammes espace-temps associés à  $c'$  et  $c''$ . Les antécédents de  $\langle x, t_0+1 \rangle$  sont les sites  $\langle x+y, t_0 \rangle$  avec  $-r \leq y \leq r$ . Comme  $h_d(t_0+1) = h_d(t_0) - r$ , on a  $x+r \leq h_d(t_0)$ . Donc soit  $\langle x+y, t_0 \rangle \in \mathfrak{C}(uvu)$ , soit  $x+y \leq h(t_0) + x_0$ . Dans le premier cas,  $\mathcal{A}^{t_0}(c')_{x+y} = \mathcal{A}^{t_0}(c'')_{x+y}$  et dans le second cas aussi puisque  $uvu$  est bloquant de direction  $h + x_0$ .

Comme tous les antécédents sont dans le même état dans les deux diagrammes, nécessairement  $\mathcal{A}^{t_0+1}(c')_x = \mathcal{A}^{t_0+1}(c'')_x$ . Et finalement  $\mathcal{A}^{t_0+1}(c)_x = \mathcal{A}^{t_0+1}(c')_x$ . Donc  $\langle x, t_0+1 \rangle \in \mathfrak{C}(u)$ .

Ainsi,  $uvu$  génère bien une cloison de direction  $h$  et d'épaisseur  $e$ . □

Si l'épaisseur  $e$  est suffisante, le mot  $uvu$  génère donc un mur.

Un automate étant une fonction continue, tous ses itérés le sont aussi, on peut donc définir une notion d'équicontinuité.

**Définition 3.1.4.** On dit que  $y \in X^{\mathbb{Z}}$  est un *point d'équicontinuité* selon la direction  $h$  pour la suite de fonctions  $(\mathcal{A}^i)_{i \in \mathbb{N}}$  lorsque :

$$\forall \varepsilon > 0, \exists \eta > 0, \forall z \in X^{\mathbb{Z}}, d(y, z) < \eta \Rightarrow \forall i \in \mathbb{N}, d(\sigma^{-h(i)}(\mathcal{A}^i(y)), \sigma^{-h(i)}(\mathcal{A}^i(z))) < \varepsilon$$

De manière équivalente, on dira que la suite de fonctions  $(\sigma^{h(i)} \circ \mathcal{A}^i)_{i \in \mathbb{N}}$  est équicontinue en  $y$ .

On note  $Eq^h(\mathcal{A}) \subseteq X^{\mathbb{Z}}$  l'ensemble des points d'équicontinuité de  $\mathcal{A}$  selon la direction  $h$ .

On dira alors que  $y$  est un point d'équicontinuité pour  $\mathcal{A}$  selon la direction  $h$ .

**Définition 3.1.5.** Une fonction  $h$  est dite à *variations bornées* lorsque

$$\exists M \in \mathbb{N}, \forall t \in \mathbb{N}, |h(t+1) - h(t)| \leq M$$

On note  $\mathcal{F}_b = \{h \in \mathcal{F}, \exists M \in \mathbb{N}, \forall t \in \mathbb{N}, |h(t+1) - h(t)| \leq M\}$  l'ensemble des fonctions à variations bornées.

Lorsqu'une fonction  $h$  est à variations bornées, l'existence d'un point d'équicontinuité selon  $h$  est équivalente à l'existence d'un mot bloquant. Dans ce cas, on constate aussi qu'il existe une fonction d'épaisseur constante (supérieure à  $\max_t |h(t+1) - h(t)| + r$ ) acceptable pour  $h$ .

**Proposition 3.1.3.** Soit  $\mathcal{A}$  un automate cellulaire sur l'alphabet  $X$ . Pour une fonction  $h \in \mathcal{F}_b$ , les trois propositions suivantes sont équivalentes :

1. il existe un mot bloquant de direction  $h$  pour  $\mathcal{A}$ ;
2. il existe un mot qui génère un mur de direction  $h$  et d'épaisseur  $e$  acceptable;
3. la suite  $(\mathcal{A}^k)_k$  a un point d'équicontinuité  $y \in X^{\mathbb{Z}}$  selon la direction  $h$ .

*Démonstration.* Les points 1 et 2 sont équivalents d'après les lemmes 3.1.1 et 3.1.2. On montre maintenant que 1  $\Rightarrow$  3.

Si  $u$  est bloquant de direction  $h$  pour  $\mathcal{A}$ , on va montrer que  $y = u^{\mathbb{Z}} \in [u]$  est un point d'équicontinuité de  $\mathcal{A}$  selon  $h$ . Soit  $\varepsilon > 0$ , il existe  $k > 0$  tel que  $2^{-k|u|} \leq \varepsilon$ , on pose alors  $\eta = 2^{-(k+1)|u|}$ . Soit  $c \in X^{\mathbb{Z}}$  tel que  $d(c, y) < \eta$ , alors  $c \in [u^{2k+1}]$ . Or  $u^{2k+1}$  est bloquant de directions  $h - k|u|$  et  $h + k|u|$ . Donc  $\forall x, t \in \mathbb{Z} \times \mathbb{N}$  si  $h(t) - k|u| \leq x \leq h(t) + k|u|$ , alors  $\mathcal{A}^t(y)_x = \mathcal{A}^t(c)_x$ . Donc :

$$\forall t \in \mathbb{N}, d(\sigma^{-h(t)}(\mathcal{A}^t(y)), \sigma^{-h(t)}(\mathcal{A}^t(c))) < 2^{-k|u|} \leq \varepsilon$$

Pour conclure, il reste à montrer que 3  $\Rightarrow$  2. Soit  $y$  un point d'équicontinuité pour  $\mathcal{A}$  selon  $h$ . Soit  $e$  une épaisseur constante égale à  $M$  acceptable pour un mur de direction  $h$  (telle que  $\forall t \in \mathbb{N}, e(t) = M$  et  $\max_t |h(t+1) - h(t)| \leq M - r$ ).

On prend alors  $\varepsilon > 0$  tel que  $-2\log_2 \varepsilon + 1 \geq M$ . Comme  $y$  est un point d'équicontinuité, on en déduit un  $\eta > 0$  tel que  $\forall z \in X^{\mathbb{Z}}, d(y, z) < \eta \Rightarrow \forall i \in \mathbb{N}, d(\sigma^{-h(i)}(\mathcal{A}^i(y)), \sigma^{-h(i)}(\mathcal{A}^i(z))) < \varepsilon$ . On prend alors  $N = -\log_2 \eta$  et  $u = y_{[-N, N]}$ . Dans ce cas, la propriété se réécrit en

$$\forall z \in [u], \forall k \in \mathbb{Z} \text{ tel que } -\frac{M}{2} \leq k \leq \frac{M}{2}, \forall t \in \mathbb{N}, \langle h(t) + k, t \rangle \in \mathfrak{C}(u)$$

Et  $u$  génère un mur d'épaisseur  $e$  et de direction  $h$ . □

*Remarque 3.1.1.* Pour une fonction à variations non bornées, seule l'implication  $3 \Rightarrow 2$  (et donc aussi  $3 \Rightarrow 1$ ) est remise en cause. Toutefois, en l'absence de contre exemple, cela reste une question ouverte.

**Corollaire 3.1.4.** *Soit  $\mathcal{A}$  un automate cellulaire sur l'alphabet  $X$ . Pour une fonction  $h \in \mathcal{F}$ , les deux propositions suivantes sont équivalentes :*

1. *il existe un mot bloquant de direction  $h$  pour  $\mathcal{A}$ ;*
2. *il existe un mot qui génère un mur de direction  $h$  et d'épaisseur  $e$  acceptable.*

*De plus, elles impliquent que la suite  $(\mathcal{A}^k)_k$  a un point d'équicontinuité  $y \in X^{\mathbb{Z}}$  selon la direction  $h$ .*

*Remarque 3.1.2.* On prouve de la même manière que, s'il existe un mot bloquant  $u$  pour un AC, alors pour tout mot  $v$  la configuration  ${}^{\mathbb{N}}uvu^{\mathbb{N}}$  est un point d'équicontinuité. Il est donc aisé d'obtenir un point d'équicontinuité arbitrairement proche d'une configuration donnée dans ce cas.  $Bl^h(\mathcal{A})$  est donc vide ou dense dans l'ensemble des configurations (de même pour  $Eq^h(\mathcal{A})$  si  $h$  est à variations bornées).

## 3.2 Dynamiques

On va ici transposer les définitions classiques de dynamique des automates cellulaires dans le cas directionnel. On reprend donc les définitions de sensibilité, équicontinuité et expansivité [22] pour les généraliser à une direction quelconque.

### 3.2.1 Équicontinuité

**Définition 3.2.1.** Un automate cellulaire  $\mathcal{A}$  est *bloquant* selon  $h \in \mathcal{F}$  lorsque  $Bl^h(\mathcal{A}) = X^{\mathbb{Z}}$ , c'est à dire que pour toute configuration  $x \in X^{\mathbb{Z}}$ , il existe  $l \in \mathbb{N}$  tel que  $x_{[-l, l]}$  est bloquant de direction  $h$ .

Lorsque  $h$  est à variations bornées, on verra que cette définition rejoint la définition classique d'équicontinuité :

**Définition 3.2.2.** Un automate cellulaire  $\mathcal{A}$  est *équicontinu* selon  $h \in \mathcal{F}$  si et seulement si  $Eq^h(\mathcal{A}) = X^{\mathbb{Z}}$ , ou de manière équivalente :

$$\forall x \in X^{\mathbb{Z}}, \forall \varepsilon > 0, \exists \eta > 0,$$

$$\forall i \in \mathbb{N}, \forall y \in X^{\mathbb{Z}}, d(x, y) < \eta \Rightarrow d(\sigma^{-h(i)}(\mathcal{A}^i(x)), \sigma^{-h(i)}(\mathcal{A}^i(y))) < \varepsilon$$

Ici, pour un automate cellulaire bloquant, tout mot assez grand au centre d'une configuration est un mot bloquant de direction  $h$ . Cependant, la taille de ce mot n'est pas a priori bornée. On va voir qu'en fait cette taille est nécessairement bornée, c'est à dire que l'on peut effectuer une inversion de quantificateurs dans la définition d'un automate bloquant, et aussi dans celle d'automate équicontinu :

**Proposition 3.2.1.** *Un automate cellulaire  $\mathcal{A}$  est équicontinu selon  $h \in \mathcal{F}$  si et seulement si  $\forall \varepsilon > 0, \exists \eta > 0,$*

$$\forall i \in \mathbb{N}, \forall x \in X^{\mathbb{Z}}, \forall y \in X^{\mathbb{Z}}, d(x, y) < \eta \Rightarrow d(\sigma^{-h(i)}(\mathcal{A}^i(x)), \sigma^{-h(i)}(\mathcal{A}^i(y))) < \varepsilon$$

*Démonstration.* Il suffit de montrer que si un AC est équicontinu de direction  $h$ , il a cette propriété, la réciproque est claire. On va donc supposer que  $\mathcal{A}$  est équicontinu. Soit  $\varepsilon > 0$ , à toute configuration  $x$  on associe  $\eta_x > 0$  tel que  $\forall y \in B(x, \eta_x),$

$$\forall i \in \mathbb{N}, d(\sigma^{-h(i)}(\mathcal{A}^i(x)), \sigma^{-h(i)}(\mathcal{A}^i(y))) < \varepsilon$$

Or  $X^{\mathbb{Z}} \subseteq \bigcup_{x \in X^{\mathbb{Z}}} B(x, \eta_x)$  et comme  $X^{\mathbb{Z}}$  est compact, on peut extraire un recouvrement fini : il existe  $x_1, \dots, x_j$  ( $j \in \mathbb{N}$ ) tels que  $X^{\mathbb{Z}} \subseteq \bigcup_{1 \leq i \leq j} B(x_i, \eta_{x_i})$ .

On note maintenant  $\eta = \min_{1 \leq i \leq j} \eta_{x_i}$ . Pour toute configuration  $x \in X^{\mathbb{Z}}$ , il existe donc  $1 \leq i \leq j$  tel que  $x \in B(x_i, \eta)$ . Et pour tout  $y \in B(x, \eta)$ , on a  $y \in B(x_i, \eta)$ . Donc

$$\forall t \in \mathbb{N}, d(\sigma^{-h(t)}(\mathcal{A}^t(x)), \sigma^{-h(t)}(\mathcal{A}^t(x_i))) < \varepsilon$$

et

$$\forall t \in \mathbb{N}, d(\sigma^{-h(t)}(\mathcal{A}^t(y)), \sigma^{-h(t)}(\mathcal{A}^t(x_i))) < \varepsilon$$

Finalement  $\forall t \in \mathbb{N}, d(\sigma^{-h(t)}(\mathcal{A}^t(y)), \sigma^{-h(t)}(\mathcal{A}^t(y))) < \varepsilon.$   $\square$

De la même manière, on peut prouver la proposition suivante :

**Proposition 3.2.2.** *Un automate cellulaire  $\mathcal{A}$  est bloquant selon  $h \in \mathcal{F}$  si et seulement si il existe  $l \in \mathbb{N}$  tel que pour toute configuration  $x \in X^{\mathbb{Z}}, x_{[-l, l]}$  est bloquant de direction  $h$ .*

*Démonstration.* Pour l'implication réciproque, on suppose le contraire. Il existe alors une suite de mots de tailles croissantes qui ne sont pas bloquants. Grâce à cette suite, on peut donc extraire une configuration dont aucun sous-mot n'est bloquant. Cela revient à contredire le fait que  $\mathcal{A}$  soit bloquant.  $\square$

Il existe donc une taille minimale de mots telle que tout mot de cette taille fixe l'ensemble des sites  $\{(h(t), t), t \in \mathbb{N}\}$ .

Si on a seulement  $\emptyset \subsetneq Bl^h(\mathcal{A}) \subsetneq X^{\mathbb{Z}}$ , il existe des mots bloquants de direction  $h$ , mais il existe aussi des mots non bloquants de taille arbitrairement grande.

Finalement, on peut montrer qu'être bloquant implique l'équicontinuité et que ces deux notions sont équivalentes lorsque l'on traite des fonctions à variations bornées :

**Proposition 3.2.3.** *Pour  $h \in \mathcal{F}$  et  $\mathcal{A}$  un automate cellulaire :*

$$\mathcal{A} \text{ est bloquant} \Rightarrow \mathcal{A} \text{ est équicontinu}$$

*Pour  $h \in \mathcal{F}_b$  et  $\mathcal{A}$  un automate cellulaire :*

$$\mathcal{A} \text{ est bloquant} \Leftrightarrow \mathcal{A} \text{ est équicontinu}$$

*Démonstration.* Si un automate cellulaire est bloquant, d'après la proposition 3.2.2, tout mot de taille  $2l + 1$  est bloquant pour  $l \in \mathbb{N}$  donné. Ainsi, toute configuration peut être vue comme une succession de mots bloquants, donc on vérifie la propriété d'équicontinuité.

Si un automate cellulaire est équicontinu, la proposition 3.2.1 dit que pour tout  $\varepsilon > 0$ , tout mot de taille  $-2 \log_2(\eta)$  pour un certain  $\eta > 0$  génère une cloison d'épaisseur  $-2 \log_2(\varepsilon)$ . Comme  $h$  est à variations bornées, la fonction constante égale à  $(\max_t (h(t+1) - h(t) + r))$  est une épaisseur acceptable constante. Ainsi, en prenant  $\varepsilon$  de sorte que  $-2 \log_2(\varepsilon)$  soit cette épaisseur acceptable, tous les mots de taille  $-2 \log_2(\eta)$  sont bloquants, et  $\mathcal{A}$  est bloquant.  $\square$

### 3.2.2 Sensibilité et expansivité

On se place maintenant dans le cas où  $Bl^h(\mathcal{A}) = \emptyset$ . L'absence de mots bloquants de direction  $h$  signifie qu'on ne peut isoler un sous ensemble de cellules. C'est la propriété de *sensibilité* qui est la négation de l'existence d'un point d'équicontinuité :

**Définition 3.2.3.** Un automate cellulaire  $\mathcal{A}$  est sensible selon  $h$  lorsque  $Bl^h(\mathcal{A}) = \emptyset$ , ou de manière équivalente :

$$\forall x \in X^{\mathbb{Z}}, \exists \varepsilon > 0, \forall \eta > 0, \exists y \in B(x, \eta), \exists i \in \mathbb{N}, d(\sigma^{-h(i)}(\mathcal{A}^i(x)), \sigma^{-h(i)}(\mathcal{A}^i(y))) \geq \varepsilon.$$

Lorsqu'un automate est sensible, pour toute configuration il est possible de trouver une modification de cette configuration arbitrairement loin telle que cette modification entraîne à son tour une différence dans le diagramme espace-temps le long de  $h$ .

Une propriété plus forte consiste à requérir que toute modification d'une configuration finisse par se répercuter le long de  $h$ , c'est la propriété d'*expansivité* :

**Définition 3.2.4.** Un automate cellulaire  $\mathcal{A}$  est expansif selon la direction  $h$  lorsque :

$$\exists \varepsilon > 0, \forall x \in X^{\mathbb{Z}}, \forall y \neq x, \exists i \in \mathbb{N}, d(\sigma^{-h(i)}(\mathcal{A}^i(x)), \sigma^{-h(i)}(\mathcal{A}^i(y))) \geq \varepsilon.$$

### 3.2.3 Classification

On peut maintenant donner une classification des automates cellulaires similaire à la classification de Kůrka pour le cas de la direction verticale [22] :

**Théorème 3.2.4.** *Pour toute fonction  $h \in \mathcal{F}$ , chaque AC  $\mathcal{A}$  correspond à exactement un des cas suivants :*

1.  $Eq^h(\mathcal{A}) = X^{\mathbb{Z}}$ ,  $\mathcal{A}$  est équicontinu selon la direction  $h$  ;
2.  $\emptyset \subsetneq Eq^h(\mathcal{A}) \subsetneq X^{\mathbb{Z}}$ ,  $\mathcal{A}$  a des points d'équicontinuité selon la direction  $h$ , mais n'est pas équicontinu ;
3.  $\mathcal{A}$  est sensible selon la direction  $h$ , mais pas expansif (donc  $Eq^h(\mathcal{A}) = \emptyset$ ) ;
4.  $\mathcal{A}$  est expansif selon la direction  $h$  (donc  $Eq^h(\mathcal{A}) = \emptyset$ ).

Toutefois, lorsque les fonctions sont à variations bornées, l'existence d'un point d'équicontinuité correspond à la capacité d'un mot à arrêter tous les flux d'informations, et donc la sensibilité correspond à la possibilité d'envoyer au moins une information arbitrairement loin dans tout contexte. Dans le cas de fonctions à variations non bornées, des informations peuvent a priori traverser la fonction selon laquelle une configuration est un point d'équicontinuité. On peut donc aussi proposer la classification suivante :

**Théorème 3.2.5.** *Pour toute fonction  $h \in \mathcal{F}$ , chaque AC  $\mathcal{A}$  correspond à exactement un des cas suivants :*

1.  $Bl^h(\mathcal{A}) = X^{\mathbb{Z}}$ ,  $\mathcal{A}$  est bloquant selon la direction  $h$  ;
2.  $\emptyset \subsetneq Bl^h(\mathcal{A}) \subsetneq X^{\mathbb{Z}}$ , il existe des mots bloquants de direction  $h$ , mais  $\mathcal{A}$  n'est pas bloquant de direction  $h$  ;
3.  $\emptyset \subsetneq Eq^h(\mathcal{A}) \subseteq X^{\mathbb{Z}}$  et  $Bl^h(\mathcal{A}) = \emptyset$ ,  $\mathcal{A}$  a des points d'équicontinuité selon la direction  $h$  mais pas de mots bloquants ;
4.  $\mathcal{A}$  est sensible selon la direction  $h$ , mais pas expansif (donc  $Eq^h(\mathcal{A}) = \emptyset$ ) ;
5.  $\mathcal{A}$  est expansif selon la direction  $h$  (donc  $Eq^h(\mathcal{A}) = \emptyset$ ).

Mais la classe 3 de cette classification peut être vide. Si l'existence d'un point d'équicontinuité, implique l'existence d'un mot bloquant, cette classe est vide. La place d'un automate cellulaire équicontinu n'est pas évidente non plus, il sera nécessairement dans une des trois premières classes mais on ne peut pas affirmer ou infirmer pour le moment qu'être équicontinu revient à être bloquant.

### 3.3 Ensembles de directions

Il est donc possible, en fixant une direction, de caractériser la dynamique d'un automate cellulaire selon cette direction de la même manière que dans le cas particulier de la direction verticale. Mais en considérant différentes directions, il est aussi intéressant de regarder l'ensemble des directions selon lesquelles on observe une certaine dynamique, en particulier l'ensemble des directions selon lesquelles il existe un mot bloquant nous renseigne sur les flux d'informations dans l'automate.

#### 3.3.1 Ordre sur les fonctions

On commence par préciser la structure de l'ensemble des fonctions. En effet, par exemple, deux fonctions « proches » ne devront pas être différenciées. Les fonctions regardées sont toujours dans l'ensemble  $\mathcal{F} = \{h : \mathbb{N} \mapsto \mathbb{Z}\}$  ou  $\mathcal{F} = \{h : \mathbb{Z} \mapsto \mathbb{Z}\}$  pour des automates cellulaires réversibles.

En particulier, pour  $\alpha \in \mathbb{R}$ , on va noter  $\Delta_\alpha : t \in \mathbb{N} \mapsto \lfloor \alpha t \rfloor$ , la fonction dont la représentation est la droite discrète de pente  $\alpha$ . Cette représentation est périodique si et seulement si  $\alpha \in \mathbb{Q}$ . Un signal de vitesse  $\alpha$ , comme défini en 1.3.1, décrit la droite discrète de pente  $\alpha$ .

On définit maintenant un ordre sur ces fonctions :

**Définition 3.3.1.** Pour  $h_1, h_2 \in \mathcal{F}$ , on note  $h_1 \preceq h_2$  lorsqu'il existe  $M \in \mathbb{N}$  tel que  $\forall t \in \mathbb{N}, h_1(t) \leq h_2(t) + M$ .

Si  $h_1 \preceq h_2$  et  $h_2 \preceq h_1$ , alors on note  $h_1 \sim h_2$ .

Enfin on note  $h_1 \prec h_2$  lorsque  $h_1 \preceq h_2$  mais  $h_1 \not\sim h_2$ .

Clairement  $\sim$  est une relation d'équivalence et  $\preceq$  est transitive. On s'intéresse à l'ordre engendré sur  $\mathcal{F}$  mais aussi sur  $\mathcal{F}_b$  et plus tard sur l'ensemble des droites.

#### 3.3.2 Ensemble de directions bloquantes

On définit maintenant deux ensembles de directions selon lesquelles un AC a des mots bloquants et selon lesquelles un AC est bloquant :

**Définition 3.3.2.** Pour un automate cellulaire  $\mathcal{A}$ , on note :

$$\begin{aligned} \mathbf{A}(\mathcal{A}) &= \{h \in \mathcal{F}, Bl^h(\mathcal{A}) \neq \emptyset\} \\ \mathbf{A}'(\mathcal{A}) &= \{h \in \mathcal{F}, Bl^h(\mathcal{A}) = X^{\mathbb{Z}}\} \end{aligned}$$

La notion d'équivalence sur les fonctions est adaptée à ces définitions puisqu'un mot bloquant selon une direction permet d'en construire un (qui est une puissance du premier) selon toute direction équivalente. On a donc la proposition suivante :

**Proposition 3.3.1.**

- $\forall h, h' \in \mathcal{F}$ , si  $h \sim h'$  alors  $h \in \mathbf{A}(\mathcal{A})$  (resp.  $\mathbf{A}'(\mathcal{A})$ )  $\Leftrightarrow h' \in \mathbf{A}(\mathcal{A})$  (resp.  $\mathbf{A}'(\mathcal{A})$ );
- $\mathbf{A}'(\mathcal{A}) \subseteq \mathbf{A}(\mathcal{A})$ .

**3.3.3 Propriétés de  $\mathbf{A}(\mathcal{A})$** 

L'ensemble des directions selon lesquelles un automate cellulaire a un mot bloquant permet de donner une famille de mots bloquants représentant les directions. On va montrer qu'en fait, tout mot bloquant l'est selon chaque direction de cet ensemble, modulo l'équivalence sur les fonctions définies plus haut.

**Proposition 3.3.2.** *Soit  $\mathcal{A}$  un automate cellulaire, et  $u$  un mot bloquant de direction  $h \in \mathcal{F}$  pour  $\mathcal{A}$ , alors pour toute direction  $g \in \mathbf{A}(\mathcal{A})$ , il existe  $h' \sim g$  tel que  $u$  soit bloquant de direction  $h'$ .*

*Démonstration.* Supposons que  $v$  soit bloquant de direction  $g$ , on note  $k = \frac{1}{2}(|u| + |v|)$ . On définit la fonction  $h' : \mathbb{N} \mapsto \mathbb{Z}$  par :

- $h'(x) = g(x) + k$  si  $g(x) + k \leq h(x)$  ;
- $h'(x) = g(x) - k$  si  $g(x) - k \geq h(x)$  ;
- $h'(x) = h(x)$  sinon.

Les courbes des fonctions  $g - k$  et  $g + k$  sont donc des translations de la courbe de  $g$  ayant pour origines  $g(0) - k$  et  $g(0) + k$ . Comme illustré par la figure 3.2,  $h'$  est égale à  $g + k$  ou  $g - k$  sauf lorsque  $h$  est entre ces deux fonctions, ou dit autrement  $h'$  est la fonction au milieu parmi  $h$ ,  $g - k$  et  $g + k$ . Cela implique que  $h' \sim g$ .

On va montrer que  $u$  est bloquant de direction  $h'$ . Soient deux configurations  $c, c' \in [u]$  telles que  $c_y = c'_y$  lorsque  $y > 0$ . Soit  $t \in \mathbb{N}$  et  $x \geq h'(t)$ . Si  $x \geq h(t)$ , par définition, comme  $u$  est bloquant de direction  $h$ , on a  $\mathcal{A}^t(c)_x = \mathcal{A}^t(c')_x$ . Supposons maintenant que  $g(t) + k \leq x \leq h(t)$ .

On va utiliser le fait que  $v$  est bloquant. Soit  $a$  un état de  $X$ , on prend la configuration  $c_1 = c_{]-\infty, -\lfloor \frac{|u|}{2} \rfloor - 1]} uva^{\mathbb{N}}$ . Comme  $u$  est bloquant de direction  $h$ ,  $\mathcal{A}^t(c)_x = \mathcal{A}^t(c_1)_x$ . De la même manière, on construit une configuration  $c'_1 = c'_{]-\infty, -\lfloor \frac{|u|}{2} \rfloor - 1]} uva^{\mathbb{N}}$  telle que  $\mathcal{A}^t(c')_x = \mathcal{A}^t(c'_1)_x$ .

$\forall y > k$ ,  $(c_1)_y = (c'_1)_y$  et comme  $v$  est bloquant de direction  $g$ , et qu'il est décalé de  $k$  dans  $c_1$  et  $c'_1$ , on peut aussi affirmer que  $\mathcal{A}^t(c_1)_x = \mathcal{A}^t(c'_1)_x$  puisque  $g(t) + k \leq x$ . Finalement  $\mathcal{A}^t(c)_x = \mathcal{A}^t(c_1)_x = \mathcal{A}^t(c'_1)_x = \mathcal{A}^t(c')_x$ .

On prouve la deuxième partie ( $x \geq h'(t)$ ) de manière symétrique en utilisant des configurations de  $[vu]_{-|v| - \lfloor \frac{|u|}{2} \rfloor}$  et  $u$  est bloquant de direction  $h'$ .

□

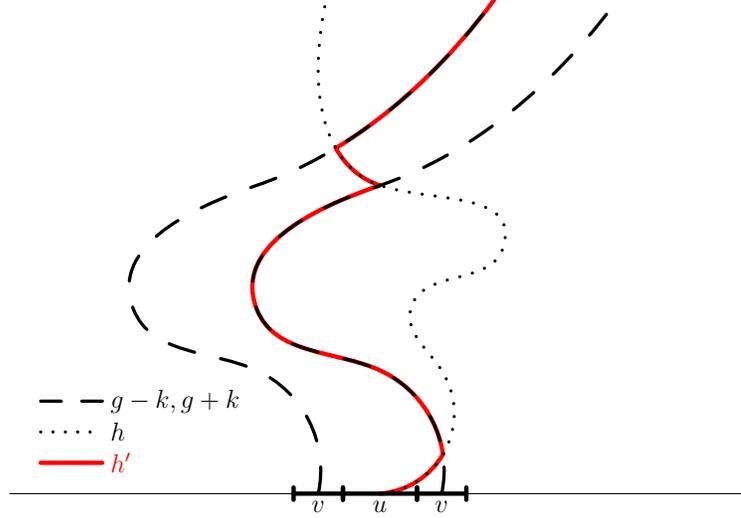


FIGURE 3.2 –

Dans le cas où un mot est bloquant selon deux directions non équivalentes, tout site situé entre les représentations des deux fonctions de direction ne peut être influencé ni par la demi-configuration à gauche du mot bloquant, ni par celle à droite du mot bloquant. Ces sites sont donc dans l'ensemble de conséquences du mot :

**Lemme 3.3.3.** *Soit  $u$  bloquant pour l'automate cellulaire  $\mathcal{A}$  dans les directions  $h \in \mathcal{F}$  et  $h' \in \mathcal{F}$ , alors  $\forall t \in \mathbb{N}$ , si  $h(t) \leq x \leq h'(t)$  ou si  $h'(t) \leq x \leq h(t)$ ,  $\langle x, t \rangle \in \mathfrak{C}(u)$ .*

*Démonstration.* Soit  $c$  et  $c'$  deux configurations de  $[u]$ . Soit  $t \in \mathbb{N}$  et prenons  $h(t) \leq x \leq h'(t)$ .

Soit  $c'' \in [u]$  telle que  $\forall y > 0, c''_y = c_y$  et  $\forall y < 0, c''_y = c'_y$ . Comme  $u$  est bloquant selon  $h$  et  $x \geq h(t)$ , on a  $\mathcal{A}^t(c)_x = \mathcal{A}^t(c'')_x$ . Et comme  $u$  est bloquant selon  $h'$  et  $x \leq h'(t)$ , on a  $\mathcal{A}^t(c')_x = \mathcal{A}^t(c'')_x$ . Donc  $\mathcal{A}^t(c)_x = \mathcal{A}^t(c')_x$  et  $\langle x, t \rangle \in \mathfrak{C}(u)$ .  $\square$

En particulier, le lemme précédent montre que si un mot est bloquant selon deux directions, il l'est aussi selon toute direction intermédiaire. Ce résultat n'est pas a priori vrai pour des directions d'équicontinuité qui ne sont pas bloquantes. On va maintenant montrer qu'il est possible de construire des directions bloquantes minimales et maximales pour tout mot, modulo l'équivalence sur les fonctions.

**Proposition 3.3.4.** *Soit  $\mathcal{A}$  un automate cellulaire, alors il existe deux fonctions calculables  $h_{min}, h_{max} \in \mathcal{F}$ , telles que  $\mathbf{A}(\mathcal{A}) = \{h \in \mathcal{F}, h_{min} \preceq h \preceq h_{max}\}$ .*

*Démonstration.* Si il n'existe de mot bloquant selon aucune direction, il n'y a rien à prouver (on obtient  $\mathbf{A}(\mathcal{A}) = \emptyset$  avec par exemple  $h_{min} = \Delta_1$  et  $h_{max} = \Delta_0$ ). On suppose donc qu'il existe un mot bloquant  $v$  de direction  $g \in \mathcal{F}$ . Alors  $u = vv$  génère un mur d'épaisseur  $|v|$  et en particulier,  $\forall t \in \mathbb{N}, \langle g(t), t \rangle \in \mathfrak{C}(u)$ . On construit  $h_{min}$  et  $h_{max}$  comme suit :

- $h_{min}(0) = g(0) = h_{max}(0)$  ;
- $\forall t \in \mathbb{N}, h_{min}(t) = \min\{x \in \mathbb{Z}, \langle x, t \rangle \in \mathfrak{C}(u)\}$
- $\forall t \in \mathbb{N}, h_{max}(t) = \max\{x \in \mathbb{Z}, \langle x, t \rangle \in \mathfrak{C}(u)\}$

On a donc nécessairement : pour tout  $t \in \mathbb{N}$ ,  $h_{min}(t) \leq g(t) \leq h_{max}(t)$ . Et de même pour toute direction  $h$  selon laquelle  $v$  est bloquant. On construit maintenant  $h_g \sim h_{min}$  et  $h_d \sim h_{max}$  :

- $\forall t \in \mathbb{N}, h_g(t) = \min(h_{min}(t) + |u|, g(t))$
- $\forall t \in \mathbb{N}, h_d(t) = \max(h_{max}(t) - |u|, g(t))$

On montre ensuite que  $u$  est bloquant de direction  $h_g$ , la démonstration est symétrique pour  $h_d$ . On commence par montrer que pour tout  $t \in \mathbb{N}^*$ , et  $\forall x \in \mathbb{Z}$  tel que  $h_g(t) \leq x < g(t)$ , on a  $\langle x, t \rangle \in \mathfrak{C}(u)$ . Comme  $u$  est bloquant de direction  $g$ , cela prouvera le résultat. Soit  $t \in \mathbb{N}^*$ , si  $h_g(t) = g(t)$ , il n'y a rien à montrer, sinon,  $h_{min}(t) + |u| < g(t)$ .

Soit  $k < g(t) - h_g(t)$ , on note  $w = ua^{k'}u$  pour une lettre  $a \in X$  de manière à ce que le mot  $w$  soit bloquant pour les directions  $(g - |u| - k)$  et  $(g + |u| + k)$ . Comme  $h_{min}(t) < g(t) - |u| - k$  et que la première occurrence de  $u$  crée un mur de direction  $(g - |u| - k)$ ,  $\langle h_{min}(t), t \rangle$  ne dépend pas de la seconde occurrence de  $u$ . Ce site se trouve néanmoins dans les conséquences de la seconde occurrence de  $u$  qui sont les mêmes que celles de  $u$  décalées. Il faut donc que  $\langle h_{min}(t), t \rangle$  soit dans les conséquences de la première occurrence de  $u$ . Ce qui signifie que  $\langle h_{min}(t) + |u| + k, t \rangle \in \mathfrak{C}(u)$ . Donc,  $\forall h_g(t) \leq x < g(t), \langle x, t \rangle \in \mathfrak{C}(u)$ , ce qui permet de conclure que  $u$  est bloquant selon  $h_g$ .

Pour finir la démonstration, on prend  $h \in \mathbf{A}(\mathcal{A})$ , alors d'après la proposition 3.3.2 il existe  $h' \sim h$  telle que  $v$  soit bloquant de direction  $h'$ . Grâce à ce qui précède, on peut affirmer que  $h_{min} \preceq h' \preceq h_{max}$ .  $\square$

*Remarque 3.3.1.* Les fonctions  $h_{min}$  et  $h_{max}$  sont calculables étant donné l'automate  $\mathcal{A}$  et un mot bloquant  $u$ , mais d'après la proposition 2.1.4, il n'est pas possible de décider si un mot est bloquant. Donc ces fonctions ne sont pas calculables avec comme seule entrée l'automate cellulaire.

On a donc montré que l'ensemble  $A(\mathcal{A})$  est exactement le segment (bornes incluses) entre deux fonctions de  $\mathcal{F}$ . Si l'on se restreint à  $\mathcal{F}_b$ , les trois résultats précédents restent valables, mais pour la proposition 3.3.4, les bornes  $h_{min}$  et  $h_{max}$  ne sont pas nécessairement dans  $\mathcal{F}_b$ , donc il reste à traiter le cas de  $\mathbf{A}(\mathcal{A}) \cap \mathcal{F}_b$ . Il est possible d'approcher les fonctions  $h_{min}$  et  $h_{max}$  par des fonctions à variations bornées

en utilisant des bornes de plus en plus grandes, donc  $\mathbf{A}(\mathcal{A}) \cap \mathcal{F}_b$  peut être borné par des fonctions à variations non bornées. Dans ce cas, les bornes ne sont pas incluses.

Avoir une direction de mot bloquant revient à pouvoir contraindre l'information selon cette direction, or on ne peut envoyer de l'information plus rapidement que le rayon ne le permet, donc on va prouver que les droites de vitesses le rayon et son opposé sont des bornes pour les directions de mots bloquants. Si une direction dépasse une de ces bornes, l'automate est nécessairement nilpotent.

**Proposition 3.3.5.** *Soit  $\mathcal{A}$  un automate cellulaire de rayon  $r$ . Soit  $h \in \mathcal{F}$  une direction de mot bloquant pour  $\mathcal{A}$ , si  $\mathcal{A}$  n'est pas nilpotent, alors  $\Delta_{-r} \preceq h \preceq \Delta_r$ .*

*Démonstration.* Soit  $v$  un mot bloquant pour la direction  $h$ .  $u = vv$  est un mot bloquant de direction  $h$  tel que  $\forall t \in \mathbb{N}, \langle h(t), t \rangle \in \mathfrak{C}(u)$ . On note  $k = |u|$ . Si par exemple  $h \not\preceq \Delta_r$  n'est pas vérifiée, il existe  $t \in \mathbb{N}$  tel que  $\Delta_r(t) + k < h(t)$ . Comme  $\langle h(t), t \rangle \in \mathfrak{C}(u)$ , il existe  $a \in X$  tel que  $\forall c \in [u], \mathcal{A}^t(c)_{h(t)} = a$ .

Pour tout mot  $w \in X^{2tr+1}$ , il existe  $c_w \in [u]$  telle que

$$(c_w)_{[h(t)-tr; h(t)+tr]} = w \text{ et } (\sigma^{h(t)} \circ \mathcal{A}^t(c_w))_0 = a$$

Or nécessairement  $\langle 0, t \rangle \in \mathfrak{C}((c_w)_{[-tr, tr]}) = \mathfrak{C}(w)$ . Donc,

$$\forall w \in X^{2tr+1}, \forall c \in [w], \mathcal{A}^t(c)_0 = a$$

Et  $\mathcal{A}$  est nilpotent. □

On montre de la même manière que, pour un automate cellulaire nilpotent, l'ensemble des directions de mots bloquants est exactement  $\mathcal{F}$ .

### 3.3.4 Propriétés de $\mathbf{A}'(\mathcal{A})$

On a vu que pour un AC nilpotent  $\mathbf{A}'(\mathcal{A}) = \mathcal{F}$ . On sait aussi que la propriété d'équicontinuité est stable pour la relation d'équivalence sur les fonctions. On va montrer qu'il n'existe que trois cas pour  $\mathbf{A}'(\mathcal{A})$  : c'est soit un ensemble vide, soit un ensemble plein, soit un ensemble réduit à une classe d'équivalence de fonctions. De plus, tout comme pour la proposition 3.2.1, les résultats que l'on montre ici sont valables pour un automate équicontinu, et pas nécessairement bloquant.

**Proposition 3.3.6.** *Soit  $\mathcal{A}$  un automate cellulaire et  $h, h'$  deux directions d'équicontinuité pour  $\mathcal{A}$ , alors si  $h \approx h'$ ,  $\mathcal{A}$  est nilpotent.*

*Démonstration.* D'après la proposition 3.2.1, il existe une taille telle que tout mot de cette taille ait dans ses conséquences l'ensemble  $\{(h(t), t), t \in \mathbb{N}\}$ . On note  $l_1$  cette taille. De même, il existe une taille similaire  $l_2$  pour  $h'$ . On pose  $l = \max(l_1, l_2)$ . Comme  $h \approx h'$ , on peut prendre par exemple  $t \in \mathbb{N}$  tel que  $h(t) - h'(t) \geq l$  (le cas  $h'(t) - h(t) \geq l$  est symétrique). Si  $\mathcal{A}$  n'est pas nilpotent, il existe deux états  $a, b \in X$  ( $a \neq b$ ) et deux configurations  $c, c' \in X^{\mathbb{Z}}$  telles que  $\mathcal{A}^t(c)_0 = a$  et  $\mathcal{A}^t(c')_0 = b$ .

On note alors

$$u = c_{[-h(t)-\lfloor \frac{l-1}{2} \rfloor, -h(t)+\lceil \frac{l-1}{2} \rceil]} \text{ et } v = c'_{[-h'(t)-\lfloor \frac{l-1}{2} \rfloor, -h'(t)+\lceil \frac{l-1}{2} \rceil]}.$$

On prend maintenant une configuration  $c'' \in X^{\mathbb{Z}}$  telle que

$$c''_{[-h(t)-\lfloor \frac{l-1}{2} \rfloor, -h(t)+\lceil \frac{l-1}{2} \rceil]} = u \text{ et } c''_{[-h'(t)-\lfloor \frac{l-1}{2} \rfloor, -h'(t)+\lceil \frac{l-1}{2} \rceil]} = v.$$

Ceci est possible car  $h(t) - h'(t) \geq l$ . Maintenant l'équicontinuité selon  $h$  nous dit que  $\mathcal{A}^t(c'')_0 = a$  et celle de  $h'$  que  $\mathcal{A}^t(c'')_0 = b$ . D'où  $a = b$  et il s'ensuit que  $\mathcal{A}$  est nécessairement nilpotent.  $\square$

Dans le dernier cas, pour lequel l'ensemble des directions d'équicontinuité est réduit à une classe d'équivalence de fonctions, chaque site du diagramme espace-temps ne dépend que d'un nombre fini de cellules puisqu'il n'y a qu'une direction de propagation d'information. On peut alors montrer que chaque diagramme est ultimement périodique dans la direction  $h$ , et que la direction d'équicontinuité est une droite de pente rationnelle.

**Proposition 3.3.7.** *Soit  $\mathcal{A}$  un automate cellulaire non nilpotent et  $h \in \mathcal{F}$  tels que  $\mathcal{A}'(\mathcal{A}) = \{h' \in \mathcal{F}, h' \sim h\}$ . Alors, il existe  $\alpha \in \mathbb{Q}$  tel que  $h \sim \Delta_\alpha$ . De plus,  $\mathcal{A}$  est ultimement périodique.*

*Démonstration.* Comme dans la preuve précédente, on utilise la taille  $l$  telle que tout mot de taille  $l$  fixe les sites  $\{(h(t), t), t \in \mathbb{N}\}$ . Pour tout  $t \in \mathbb{N}$ , on considère alors la fonction

$$\begin{aligned} f_t : X^l &\rightarrow X \\ u &\mapsto \mathcal{A}^t(c)_{h(t)}, c \in [u] \end{aligned}$$

L'ensemble  $\{f_t, t \in \mathbb{N}\}$  est fini, donc il existe  $t_1 \neq t_2 \in \mathbb{N}$  tels que  $f_{t_1} = f_{t_2}$ . Donc, pour toute configuration  $c \in X^{\mathbb{Z}}$ ,  $\mathcal{A}^{t_1}(c)_{h(t_1)} = \mathcal{A}^{t_2}(c)_{h(t_2)}$ .

On pose  $\alpha = \frac{h(t_2) - h(t_1)}{t_2 - t_1}$ . Pour tout  $k$  et pour un mot  $u$  assez grand, les conséquences de  $u$  contiennent les sites  $\{\langle \Delta_\alpha(t) + j, t \rangle, t \leq t_2, -k \leq j \leq k\}$ , donc elles contiennent tous les sites  $\{\langle \Delta_\alpha(t) + j, t \rangle, t \in \mathbb{N}, -k \leq j \leq k\}$ , et par conséquent  $u$  est équicontinu selon  $\Delta_\alpha$ .

La taille  $l$  étant générique, pour tout  $x \in \mathbb{Z}$  et pour toute configuration  $c \in X^{\mathbb{Z}}$ , on a  $\mathcal{A}^{t_1}(c)_{h(t_1)+x} = \mathcal{A}^{t_2}(c)_{h(t_2)+x}$ . Ainsi, tous les diagrammes espace-temps de  $\mathcal{A}$  sont bien ultimement périodiques de pré-période  $t_1$  et de période  $(h(t_2) - h(t_1), t_2 - t_1)$ .  $\square$

La preuve précédente montre aussi qu'il ne peut y avoir de point d'équicontinuité selon une autre direction, donc  $\mathbf{A}(\mathcal{A}) = \mathbf{A}'(\mathcal{A}) = \{h' \in \mathcal{F}, h' \sim h\}$ .

Ces résultats montrent l'équivalence pour un automate cellulaire entre être équicontinu de direction  $h$  et être bloquant de direction  $h$ . En effet, un automate bloquant est équicontinu (proposition 3.2.3), et un automate équicontinu est soit nilpotent, donc bloquant pour toutes les directions, soit équicontinu selon une droite de pente rationnelle. Or cette droite est à variations bornées et dans ce cas, l'équicontinuité implique que l'automate soit bloquant de direction  $h$ .

**Corollaire 3.3.8.** *Soit  $\mathcal{A}$  un automate cellulaire et  $h \in \mathcal{F}$ .  $\mathcal{A}$  est équicontinu selon  $h$  si et seulement si il est bloquant de direction  $h$ .*

### 3.3.5 Caractérisation

On peut finalement proposer une caractérisation en termes de directions bloquantes.

**Théorème 3.3.9.** *Soit  $\mathcal{A}$  un automate cellulaire de rayon  $r$ , il est dans exactement une des catégories suivantes :*

1.  $\mathbf{A}'(\mathcal{A}) = \mathcal{F} = \mathbf{A}(\mathcal{A})$ , dans ce cas  $\mathcal{A}$  est nilpotent ;
2. il existe  $\alpha \in [-r, r] \cap \mathbb{Q}$  tel que  $\mathbf{A}'(\mathcal{A}) = \{h \in \mathcal{F}, h \sim \Delta_\alpha\} = \mathbf{A}(\mathcal{A})$ , alors  $(\mathcal{A}^n \circ \sigma^{\alpha n})_{n \in \mathbb{N}}$  est ultimement périodique ;
3. il existe  $h_1 \prec h_2 \in \mathcal{F}$  avec  $\Delta_{-r} \preceq h_1$  et  $h_2 \preceq \Delta_r$ , telles que  $\mathbf{A}(\mathcal{A}) = \{h \in \mathcal{F}, h_1 \preceq h \preceq h_2\}$ , et  $\mathbf{A}'(\mathcal{A}) = \emptyset$  ;
4. il existe  $h \in \mathcal{F}$  telle que  $\Delta_{-r} \preceq h \preceq \Delta_r$  et  $\mathbf{A}(\mathcal{A}) = \{h' \in \mathcal{F}, h' \sim h\}$ , alors  $\mathbf{A}'(\mathcal{A}) = \emptyset$  ;
5.  $\mathbf{A}(\mathcal{A}) = \mathbf{A}'(\mathcal{A}) = \emptyset$ .

*Démonstration.* Les propositions 3.3.6 et 3.3.7 permettent de distinguer les cas 1 et 2 des autres. Les cas 3 et 4 résultent des propositions 3.3.2 et 3.3.4. Reste le cas 5 qui contient tous les automates cellulaires restants.  $\square$

Les classes 1 et 5 sont stables par produit cartésien. En revanche, les trois autres ne le sont pas. Le produit de deux automates de la classe 2 est en classe 5 dès que les directions des deux automates ne sont pas les mêmes. Si elles sont identiques, le produit reste en classe 2.

Le produit de deux automates de la classe 3 peut être en classe 3 si l'intersection des deux ensembles de directions de mots bloquants est non vide et contient plus qu'une seule classe d'équivalence de fonctions. Il est en classe 4 si l'intersection est réduite à une seule classe d'équivalence, et en classe 5 lorsque l'intersection est vide. Enfin, le produit de deux automates de la classe 4 est en classe 4 uniquement si les directions de mots bloquants des deux automates sont les mêmes, sinon il est en classe 5.

### Exemples

On donne ici des exemples d'automates cellulaires pour chaque classe définie par le théorème 3.3.9. Tous ces exemples se placent dans le cas de fonctions à variations bornées, l'existence d'une direction de point d'équicontinuité correspond donc à l'existence d'un mot bloquant selon cette direction. La classe 1 est la classe des nilpotents, donc par exemple un automate tel que la règle locale produit toujours le même état est dans cette classe. L'automate *identité* est un représentant de la classe 2, il est équicontinu selon la direction  $\Delta_0$ , et périodique donc ultimement périodique. Plus généralement, tous les décalages sont dans cette classe.

Pour la classe 3, on utilise l'automate  $\mathcal{A}_p$  défini pour la proposition 2.2.1. En effet, la forme des conséquences nous donne l'ensemble  $\mathbf{A}(\mathcal{A}_p)$ . On note  $p : t \mapsto \sqrt{t+1}$ , la fonction représentée par la parabole. Alors  $\mathbf{A}(\mathcal{A}_p) = \{h \in \mathcal{F}, p \preceq h \preceq \Delta_1\}$ . En faisant un produit cartésien d'automates cellulaires, on réalise l'intersection des ensembles de directions de points d'équicontinuité d'après la proposition 2.1.2, donc un exemple pour la classe 4 est donné par le produit cartésien de l'automate cellulaire précédent avec celui en section 2.3. L'intersection des conséquences est l'ensemble des sites portés par la parabole et donc l'ensemble des directions de points d'équicontinuité est  $\{h \in \mathcal{F}, h \sim p\}$ .

L'automate *XOR* est un représentant de la classe 5.

### 3.3.6 Directions d'expansivité

Comme pour l'équicontinuité, il est possible de définir l'ensemble de directions selon lesquelles un automate cellulaire est expansif :

**Définition 3.3.3.** Pour un automate cellulaire  $\mathcal{A}$ , on note :

$$\mathbf{B}(\mathcal{A}) = \{h \in \mathcal{F}, \mathcal{A} \text{ est expansif selon } h\}$$

**Proposition 3.3.10.** Soit  $\mathcal{A}$  un automate cellulaire et  $h, h' \in \mathbf{B}(\mathcal{A}) \cap \mathcal{F}_b$ . Alors  $\forall g \in \mathcal{F}_b$  telle que  $h \preceq g \preceq h'$ ,  $g \in \mathbf{B}(\mathcal{A})$ .

*Démonstration.* Supposons que  $g$  ne soit pas une direction d'expansivité, alors

$$\forall \varepsilon > 0, \exists x \neq y, \forall i \in \mathbb{N}, d\left(\sigma^{-g(i)}(\mathcal{A}^i(x)), \sigma^{-g(i)}(\mathcal{A}^i(y))\right) < \varepsilon$$

Comme  $h$  est une direction d'expansivité :

$$\exists \varepsilon_h > 0, \forall x \neq y, \exists i \in \mathbb{N}, d\left(\sigma^{-h(i)}(\mathcal{A}^i(x)), \sigma^{-h(i)}(\mathcal{A}^i(y))\right) \geq \varepsilon_h$$

De plus  $h \preceq g$  donc il existe  $k \in \mathbb{N}$  tel que  $\forall t \in \mathbb{N}, h(t) \leq g(t) + k$ . On pose

$$e = \max_t (|g(t+1) - g(t)|) + r + 1 + k - \log_2 \varepsilon_h$$

puis  $\varepsilon_g = 2^{-e}$ . Alors comme  $g$  n'est pas une direction d'expansivité, avec le paramètre  $\varepsilon_g$ , il existe  $x \neq y$  tel que

$$\forall i \in \mathbb{N}, d\left(\sigma^{-g(i)}(\mathcal{A}^i(x)), \sigma^{-g(i)}(\mathcal{A}^i(y))\right) < \varepsilon_g$$

On peut prendre deux configurations  $x \neq y$  telles que  $\forall j \geq 0, x_j = y_j$  (la preuve est symétrique pour en regardant  $h'$  lorsque les différences sont pour  $j \leq 0$ ).

La propriété sur  $g$  nous dit que tous les sites  $\langle g(t) + l, t \rangle$  avec  $-e \leq l \leq e$  ont le même état dans les diagrammes espace-temps associés à  $x$  et  $y$ . Le choix de  $e$  fait qu'aucune information ne traverse cet ensemble de sites. Or les sites

$$\{\langle h(t) + n, t \rangle, t \in \mathbb{N}, \log_2 \varepsilon_h \leq n \leq -\log_2 \varepsilon_h\}$$

ne peuvent donc être influencés par les états des sites  $\{\langle j, 0 \rangle, j \geq 0\}$ . Par conséquent, tous ces sites ont le même état, ce qui contredit l'hypothèse d'expansivité selon  $h$ .  $\square$

L'expansivité ne se comporte pas aussi bien que l'équicontinuité vis à vis du pré-ordre défini sur les fonctions. En effet, on a montré que l'ensemble des directions des mots bloquants d'un automate cellulaire est un ensemble de fonctions contenues entre deux fonctions extrémales. Pour l'expansivité, en revanche, la propriété 3.3.10 montre qu'une fonction entre deux directions d'expansivité à variations bornées en est aussi une, mais on ne peut pas trouver de fonctions extrémales à l'ensemble. Certaines fonctions non comparables peuvent parfois ne pas avoir le même statut, ce qui est illustré dans l'exemple suivant.

*Exemple 3.3.1.* On s'intéresse à l'automate identité, qui est réversible et équicontinu pour la direction  $\Delta_0$ . On pourrait s'attendre à ce qu'un automate cellulaire équicontinu ne soit expansif selon aucune direction, et pourtant, il suffit ici pour être une direction d'expansivité de passer au moins une fois par chaque cellule. Par exemple, la fonction  $h$  construite comme illustrée par la figure 3.3, c'est à dire une fonction

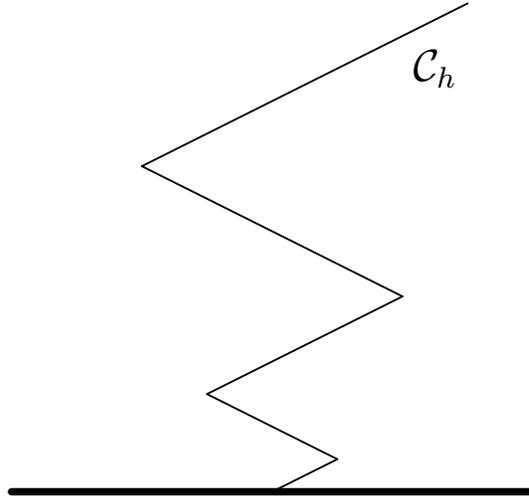


FIGURE 3.3 – La courbe  $\mathcal{C}_h$  d'une direction selon laquelle l'identité est expansive.

qui prend toutes les valeurs entières.

Bien que l'automate ne soit expansif pour aucune direction linéaire, il l'est pour  $h$ .

Une propriété naturelle est qu'une direction d'expansivité et une direction selon laquelle un mot est bloquant pour un même automate cellulaire ne peuvent pas être comparables :

**Proposition 3.3.11.** *Soit  $\mathcal{A}$  un automate cellulaire, et  $h, h' \in \mathcal{F}$ , si  $h \in \mathbf{A}(\mathcal{A})$  et  $h' \in \mathbf{B}(\mathcal{A})$ , alors  $h \not\preceq h'$  et  $h' \not\preceq h$ .*

En effet, l'expansivité dit que tout changement arbitrairement loin sur la configuration initiale finit par se répercuter autour de la courbe de  $h'$  dans le diagramme espace-temps. Or l'existence d'un mot bloquant  $u$  selon  $h$  nous dit qu'aucun changement d'une configuration initiale contenant  $u$  en son centre n'est visible autour de la courbe de  $h$ .

## 3.4 Cas particuliers

### 3.4.1 Variations non bornées

On a distingué les fonctions à variations non bornées parce que pour ces fonctions l'existence d'un point d'équicontinuité n'implique pas l'existence d'un mot bloquant.

Cependant, vis-à-vis de la caractérisation du théorème 3.3.9, il faut justifier qu'une telle fonction puisse être une borne de  $\mathbf{A}(\mathcal{A})$ . C'est l'objet de l'exemple que l'on présente dans cette partie.

### Une fonction à sauts

On va construire un automate cellulaire  $\mathcal{A}$  qui sera dans la classe 3 du théorème 3.3.9 avec  $\mathbf{A}(\mathcal{A}) = \{h' \in \mathcal{F}, \Delta_0 \preceq h' \preceq h\}$  pour la fonction  $h$  suivante :

$$\forall t \in \mathbb{N}, h(t) = \frac{k(k+1)}{2} \text{ avec } k \text{ tel que } 5 \frac{k(k+1)}{2} \leq t < 5 \frac{(k+1)(k+2)}{2}$$

Un mot  $u \in X^*$  aura pour conséquences  $\mathfrak{C}(u) = \{\langle x, t \rangle, 0 \leq x \leq h(t)\}$ . La figure 3.4 donne une représentation de cet ensemble de conséquences.

La fonction  $h$  n'est pas à variations bornées puisque  $h(t+1) - h(t) = k$  pour  $t = 5 \frac{k(k+1)}{2}$ .

### Le retour des compteurs

On va utiliser la structure de compteurs définie dans la partie 2.3. On choisit  $u = \boxed{*}$  qui initie cette construction, on détaille ensuite le calcul effectué à l'intérieur. Tout ce qui se situe entre les compteurs sera protégé, et seule une construction parfaitement identique et parallèle pourra interférer.

### Firing squad

Le *firing squad* est une construction classique d'automates cellulaires [40, 25] qui vise à synchroniser un segment de cellules. Le principe est le suivant :

- une cellule (le général) lance le processus ;
- le calcul se fait sur les cellules voisines (les soldats) ;
- après un certain temps, toutes les cellules (soldats et général) prennent simultanément le même état (feu), qu'elles n'avaient jamais pris pendant tout le calcul.

Pour l'automate  $\mathcal{A}$ , le général sera la cellule de gauche d'un segment de  $n$  cellules consécutives. Le calcul ne sortira pas de ces cellules. L'état de feu sera l'état  $\blacksquare$  qui sera l'état dans lequel seront tous les sites des conséquences de  $u(\boxed{*})$ .

Le mot  $u$  initie une construction entre les compteurs qui effectuera successivement des firing squad pour  $n \in \mathbb{N}$ .

Le firing squad peut être réalisé en temps  $2n - 1$  au minimum [40, 25], et donc en particulier en temps  $3n$ .

### L'automate

Le calcul ici décrit est illustré par la figure 3.4.

Lorsqu'une cellule contient l'état  $\blacksquare$ , elle le contient encore au temps suivant sauf si un compteur vient l'effacer. Il existe aussi un état quiescent  $\square$  qui correspond à une absence d'information et n'influence pas le calcul.

L'état  $\star$  crée un état  $\blacksquare$  et à sa droite un état  $\square$  qui sera le général. Un signal est envoyé vers la droite, parcourt tous les soldats (état  $\square$ ) puis revient jusqu'au général. Ce parcours dure  $2n$  pour  $(n-1)$  soldats. Lors de l'initialisation, il n'y a pas de soldats, donc le parcours dure 2 étapes. Ensuite le général démarre une procédure de firing squad avec tous les soldats sur sa droite. Après  $3n$  étapes supplémentaires (lors de l'étape  $N_n$ ), le général et tous les soldats sont remplacés par l'état  $\blacksquare$ .

Pour continuer le calcul, il faut qu'en parallèle du firing squad, le général et les soldats se dupliquent et que leurs doubles se placent à droite de la ligne de  $\blacksquare$  créée à l'étape  $N_n$ . De plus, un soldat est ajouté pendant le déplacement, de manière à obtenir  $n$  soldats et un général à l'étape suivante, et ainsi effectuer un firing squad avec une ligne de longueur  $(n+1)$ .

### Conséquences de $u$

Si il n'y pas d'autres occurrences de  $\star$  sur la configuration initiale, la construction se poursuit sans rencontrer d'obstacle et l'ensemble de sites dans l'état  $\blacksquare$  est bien l'ensemble de conséquences annoncé :  $\mathfrak{C}(u) = \{ \langle x, t \rangle, 0 \leq x \leq h(t) \}$ .

Si un autre état  $\star$  se trouve à gauche de  $u$  sur la configuration initiale, les compteurs vont disparaître, mais le « mur » d'états  $\blacksquare$  à l'abscisse 0 n peut pas être altéré. Si un autre état  $\star$  se trouve à droite de  $u$ , les compteurs vont encore disparaître et cette fois, la construction initiée par  $u$  va rencontrer le « mur »  $\blacksquare$  construit par l'autre état  $\star$ . L'espace entre les deux ensembles d'états  $\blacksquare$  est comblé. Une preuve similaire à celle de la proposition 2.2.1 montre que tous les états de  $\{ \langle x, t \rangle, 0 \leq x \leq h(t) \}$  sont bien  $\blacksquare$ .

Tous les autres sites peuvent prendre un état différent de  $\blacksquare$  avec la configuration initiale  $\mathbb{N}\square u\square\mathbb{N}$ . Et toutes les images successives de la configuration  $\star\mathbb{Z}$  sont  $\blacksquare\mathbb{Z}$ . Donc on a exactement  $\mathfrak{C}(u) = \{ \langle x, t \rangle, 0 \leq x \leq h(t) \}$ .

### Directions admettant un mot bloquant

On cherche maintenant les directions selon lesquelles il existe un mot bloquant pour  $\mathcal{A}$ . Clairement  $u$  est bloquant pour toutes les directions de l'ensemble  $\{h' \in \mathcal{F}, \Delta_0 \preceq h' \preceq h\}$ .

La proposition 3.3.2 montre que s'il existait une autre direction selon laquelle un mot est bloquant pour  $\mathcal{A}$ , alors  $u$  serait bloquant selon une direction équivalente. Or

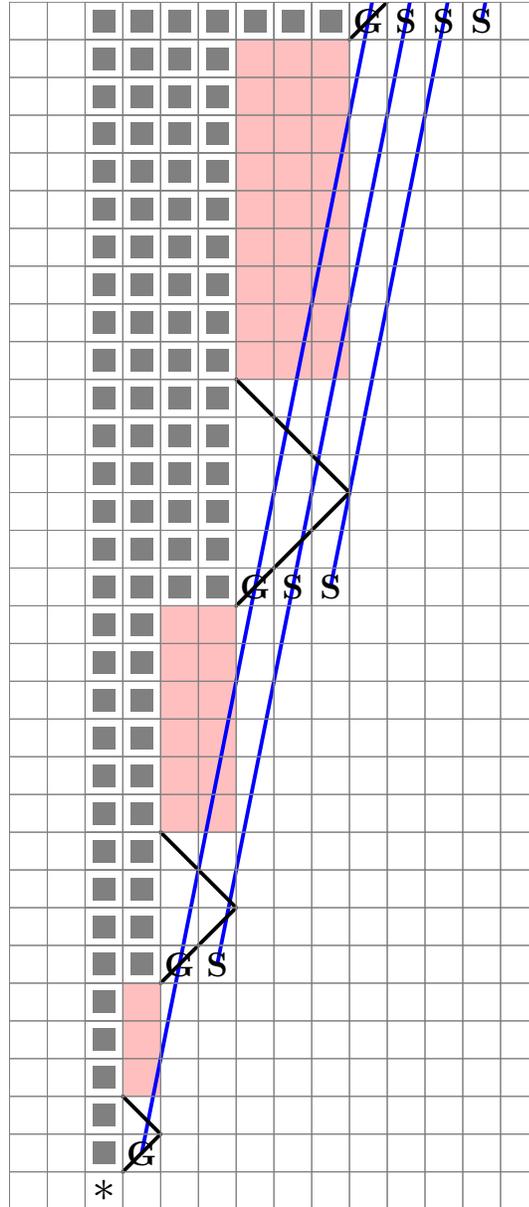


FIGURE 3.4 – La zone d'états  $\blacksquare$  est exactement l'ensemble de conséquences de  $u$ . Chaque partie  $\blacksquare$  contient une exécution du firing squad.

la preuve de la proposition 3.3.4 montre que  $\Delta_0$  et  $h$  sont bien les fonctions minimale et maximale admettant un mot bloquant pour  $\mathcal{A}$ .

On a donc bien un exemple d'automate cellulaire pour lequel une borne de l'ensemble des directions admettant un mot bloquant est une fonction à variations non bornées.

### 3.4.2 Point d'équicontinuité de direction non périodique

On s'intéresse dans cette partie au cas d'un automate cellulaire  $\mathcal{A}$  pour lequel il existe un mot bloquant selon une direction qui n'est pas une droite de pente rationnelle :  $\exists h \in \mathbf{A}(\mathcal{A})$  telle que  $\forall \alpha \in \mathbb{Q}, h \not\approx \Delta_\alpha$ . C'est le cas des automates de la classe 1 du théorème 3.3.9 (les nilpotents), mais aussi et surtout tous les automates de la classe 3, et certains de la classe 4.

On va montrer que pour un tel automate, les conséquences de tout mot bloquant sont uniformes par ligne :

**Proposition 3.4.1.** *Soit  $\mathcal{A}$  un automate cellulaire et  $h \in \mathbf{A}(\mathcal{A})$  telle que  $\forall \alpha \in \mathbb{Q}, h \not\approx \Delta_\alpha$ . Pour tout mot bloquant  $u$  de direction  $h$ , il existe une préperiode  $T \in \mathbb{N}$  telle que pour tout  $t \geq T$ , il existe un état  $a_t \in X$  tel que pour tout  $x \in \mathbb{Z}$ , si  $\langle x, t \rangle \in \mathfrak{C}(u)$  alors  $\forall c \in [u], \mathcal{A}^t(c)_x = a$ .*

*Démonstration.* Supposons que les conséquences de  $u$  ne soient pas uniformes par ligne, on choisit un mot  $u$  qui génère un mur d'épaisseur acceptable  $e > 0$ . Alors pour tout  $t \in \mathbb{N}$ ,

$$\mathcal{A}^t(u^{\mathbb{Z}}) \neq \mathcal{A}^t(\sigma(u^{\mathbb{Z}}))$$

donc il existe  $x_t$  tel que

$$\mathcal{A}^t(u^{\mathbb{Z}})_{x_t} \neq \mathcal{A}^t(\sigma(u^{\mathbb{Z}}))_{x_t}$$

Puisque  $u^{\mathbb{Z}}$  est périodique de période  $|u|$ , toutes ses images par  $\mathcal{A}$  le sont aussi, on peut donc choisir  $h(t) \leq x_t \leq h(t) + |u|$ . De plus, la suite  $(\mathcal{A}^t(u^{\mathbb{Z}}))_t$  est ultimement périodique. Soient  $m$  et  $\tau$  ses préperiode et période. Pour  $t_0 = m + \tau|u| + 1$ , on pose  $y_{t_0} = x_{t_0}$ . Soit  $r$  le rayon de  $\mathcal{A}$ , il existe nécessairement  $y_{t_0-1} \in \mathbb{N}$  tel que

$$\mathcal{A}^{t_0-1}(u^{\mathbb{Z}})_{y_{t_0-1}} \neq \mathcal{A}^{t_0-1}(\sigma(u^{\mathbb{Z}}))_{y_{t_0-1}}$$

et

$$y_{t_0} - r \leq y_{t_0-1} \leq y_{t_0} + r$$

On construit ensuite récursivement la suite  $(y_{t_0-i})_{i \leq \tau|u|}$  avec la condition

$$y_{t_0-i} - r \leq y_{t_0-i-1} \leq y_{t_0-i} + r$$

L'ensemble  $\{\langle y_t, t \rangle, m \leq t \leq t_0\}$  est de cardinal  $\tau|u| + 1$ , donc il existe  $t_1 \leq t_2$  tels que :

- $t_2 - t_1 \equiv 0[\tau]$ ;
- $y_{t_2} - y_{t_1} \equiv 0[|u|]$ .

Comme le diagramme espace-temps associé à  $u^{\mathbb{Z}}$  est périodique, on peut construire une suite périodique  $(z_t)_{t \in \mathbb{N}}$  telle que :

- $\forall t_1 \leq t \leq t_2, z_t = y_t$
- $\forall t > t_2, z_t = y_{(t \bmod t_2 - t_1)} + \lfloor \frac{t}{t_2 - t_1} \rfloor (y_{t_2} - y_{t_1})$

La suite  $(z_t)_t$  hérite de la propriété  $z_t - r \leq z_{t-1} \leq z_t + r$  pour tout  $t$  et comme  $u$  génère un mur d'épaisseur  $e > 0$ , la suite ne traverse pas les murs dirigés par  $h$  et  $h + |u|$  :

$$\forall t > t_1, h(t) \leq z_t \leq h(t) + |u|$$

On définit alors la fonction  $f : t \mapsto z_t$  (avec par exemple  $f(t) = h(t)$  pour  $t < t_1$ ). On pose  $\alpha = \frac{z_{t_2} - z_{t_1}}{t_2 - t_1} \in \mathbb{Q}$ , et clairement  $f \sim \Delta_\alpha$ . Comme de plus  $\forall t \in \mathbb{N}, |h(t) - f(t)| \leq |u|$ , on a  $f \sim h$ . Finalement,  $h \sim \Delta_\alpha$ .  $\square$

*Remarque 3.4.1.* Etant donné un mot bloquant  $u$ , tout mot de  $uX^*u$  est lui aussi bloquant. Donc pour toute configuration contenant des occurrences d'un mot bloquant arbitrairement loin sur la gauche et sur la droite, chaque site du diagramme espace-temps associé sera dans les conséquences d'un mot bloquant. La propriété précédente permet donc d'affirmer que les images d'une telle configuration tendent vers une configuration uniforme.

### 3.4.3 Automates réversibles

Les automates réversibles ne sont pas nilpotents donc ne peuvent pas être dans la classe 1 du théorème 3.3.9. Ils peuvent a priori se situer dans les quatre classes restantes, on va montrer qu'ils ne peuvent pas se trouver dans la classe 3.

**Théorème 3.4.2.** *Soit  $\mathcal{A}$  un automate cellulaire réversible, alors :*

- soit  $\mathbf{A}(\mathcal{A}) = \emptyset$ ;
- soit il existe  $\alpha \in \mathbb{Q}$  tel que  $\mathbf{A}(\mathcal{A}) = \{h \in \mathcal{F}, h \sim \Delta_\alpha\}$ .

*Démonstration.* Supposons au contraire qu'il existe  $h \in \mathbf{A}(\mathcal{A})$  telle que  $\forall \alpha \in \mathbb{Q}, h \not\sim \Delta_\alpha$ . il existe donc un mot bloquant  $u$  de direction  $h$ . On choisit  $u$  non uniforme quitte à ajouter une lettre. Alors  $u^{\mathbb{Z}}$  est une configuration périodique non uniforme et ses images aussi puisque  $\mathcal{A}$  est réversible. Or la proposition 3.4.1 dit que ses images deviennent uniformes après un certain temps.  $\square$

Les automates cellulaires réversibles sont donc en classe 2, 4 ou 5. et les seules directions acceptées en classe 4 sont les droites de pente rationnelle. Puisque les fonctions concernées sont nécessairement à variations bornées, encore une fois, existences d'un point d'équicontinuité et d'un mot bloquant sont équivalentes.

Pour un automate réversible ayant un point d'équicontinuité, on peut aussi se poser la question des temps négatifs : a t'on encore un point d'équicontinuité ? avec les mêmes mots bloquants ? On répond par l'affirmative à ces questions.

**Proposition 3.4.3.** *Soit  $\mathcal{A}$  un automate cellulaire réversible  $\Delta_\alpha \in \mathbf{A}(\mathcal{A})$  Soit  $u \in X^*$ , si  $\{\langle x, t \rangle, t \in \mathbb{N}, x = \lfloor \alpha t \rfloor\} \subseteq \mathfrak{C}(u, \mathcal{A})$ , alors  $\{\langle x, t \rangle, t \in \mathbb{Z}, x = \lfloor \alpha t \rfloor\} \subseteq \mathfrak{C}(u, \mathcal{A})$ .*

*Démonstration.* Soit  $t < 0$  et  $x = \lfloor \alpha t \rfloor$  tel que  $\langle x, t \rangle \notin \mathfrak{C}(u, \mathcal{A})$ . Il existe alors deux configurations périodiques  $c_1$  et  $c_2$  dans  $[u]$  telles que  $\mathcal{A}^t(c_1)_x \neq \mathcal{A}^t(c_2)_x$ . Comme ce sont des configurations périodiques et que  $\mathcal{A}$  est réversible, les diagrammes espace-temps associés sont périodiques, on note  $(N, T)$  une période spatio-temporelle commune. On pose  $\alpha = \frac{p}{q}$  avec  $p, q \in \mathbb{N}$ .

Il existe  $k \in \mathbb{N}$  tel que  $t' = t + kpqTN \geq 0$ . Alors  $x' = \lfloor \alpha t' \rfloor = x + kpqTN$  et comme  $(N, T)$  est une période des deux diagrammes :  $\mathcal{A}^{t'}(c_1)_{x'} = \mathcal{A}^t(c_1)_x$  et  $\mathcal{A}^{t'}(c_2)_{x'} = \mathcal{A}^t(c_2)_x$ . Or  $x' = \lfloor \alpha t' \rfloor$  donc  $\langle x', t' \rangle \in \mathfrak{C}(u, \mathcal{A})$ . Donc  $\mathcal{A}^{t'}(c_1)_{x'} = \mathcal{A}^{t'}(c_2)_{x'}$ . Finalement  $\mathcal{A}^t(c_1)_x = \mathcal{A}^t(c_2)_x$  ce qui contredit  $\langle x, t \rangle \notin \mathfrak{C}(u, \mathcal{A})$ .  $\square$

On a montré que si  $\Delta_\alpha \in \mathbf{A}(\mathcal{A})$ , alors nécessairement  $\Delta_{-\alpha} \in \mathbf{A}(\mathcal{A}^{-1})$ . Il est donc possible de définir des points d'équicontinuité sur  $\mathbb{Z}$ , auquel cas tout point d'équicontinuité selon  $\Delta_\alpha$  (resp. mot bloquant de direction  $\Delta_\alpha$ ) est un point d'équicontinuité sur  $\mathbb{Z}$  de direction  $\Delta_\alpha$  (resp. mot bloquant sur  $\mathbb{Z}$  de direction  $\Delta_\alpha$ ).

## 3.5 Restriction aux droites

Si on restreint l'ensemble des fonctions aux droites, la situation est plus simple, on obtient un ordre puisque toutes les droites sont comparables. On définit  $\tilde{\mathbf{A}}(\mathcal{A}) = \mathbf{A}(\mathcal{A}) \cap \{\Delta_\alpha, \alpha \in \mathbb{R}\}$  pour tout automate cellulaire  $\mathcal{A}$ . Comme montré par M. Sablik [37],  $\tilde{\mathbf{A}}(\mathcal{A})$  est un intervalle donné par les deux directions extrêmes, en revanche, les bornes ne sont plus nécessairement atteintes. Par exemple, l'automate défini pour la proposition 2.2.1 a des points d'équicontinuité selon toutes les droites de pente dans  $]0, 1]$ . On peut obtenir une classification similaire à celle du théorème 3.3.9, mais les propriétés des classes 3 et 4 vont changer. On commence par définir les réels récursivement énumérables qui jouent un rôle particulier vis à vis de  $\tilde{\mathbf{A}}(\mathcal{A})$ .

### 3.5.1 Réels récursivement énumérables

**Définition 3.5.1.**  $\alpha \in \mathbb{R}$  est dit *récursivement énumérable* (r.e.) lorsqu'il existe une suite calculable de rationnels qui converge vers  $\alpha$ .

On peut affiner cette définition et définir deux classes de réels r.e., que l'on peut approcher par la gauche ou par la droite.

**Définition 3.5.2.**  $\alpha \in \mathbb{R}$  est dit *récurivement énumérable à gauche* (r.e.g.) lorsqu'il existe une suite croissante calculable de rationnels qui converge vers  $\alpha$ .  
 $\alpha \in \mathbb{R}$  est dit *récurivement énumérable à droite* (r.e.d.) lorsqu'il existe une suite décroissante calculable de rationnels qui converge vers  $\alpha$ .

*Remarque 3.5.1.* Un réel à la fois r.e.g. et r.e.d. est calculable [1].

### 3.5.2 Bornes de $\tilde{\mathbf{A}}(\mathcal{A})$

On va montrer ici que les pentes des bornes de l'intervalle  $\tilde{\mathbf{A}}(\mathcal{A})$  sont nécessairement récurivement énumérables, elles peuvent être r.e.g. ou r.e.d. lorsque l'intervalle est fermé. On commence par s'intéresser à l'intervalle des pentes de droites selon lesquelles un mot donné est bloquant. Pour  $u \in X^{\mathbb{Z}}$ , on note  $I_u = \{\alpha, u \text{ est bloquant selon } \Delta_\alpha\}$ . D'après la proposition 3.3.4,  $I_u$  est un intervalle que l'on note  $[a_u, b_u]$ , il peut a priori être indifféremment ouvert ou fermé de chaque côté.

**Proposition 3.5.1.** *Soit  $\mathcal{A}$  un automate cellulaire et  $u \in X^*$ . Alors il existe  $v \in u^*$ , tel que  $I_v = [a_v, b_v]$  avec  $a_v$  r.e.g. et  $b_v$  r.e.d..*

*Démonstration.* Si  $u$  n'est bloquant selon aucune droite, il n'y a rien à prouver (on prend  $v = u$  et  $b_v < a_v$  par exemple), sinon,  $u$  est bloquant selon une droite.

$$\forall \alpha \in \mathbb{R}, \forall t \in \mathbb{N}, |\Delta_\alpha(t+1) - \Delta_\alpha(t)| \leq \alpha$$

Or  $\alpha \leq r$  si  $\mathcal{A}$  n'est pas nilpotent d'après la proposition 3.3.5, donc  $e = r$  est une épaisseur acceptable pour toute droite  $\Delta_\alpha, \alpha \in \mathbb{R}$ . Comme  $u$  est bloquant, il existe  $v \in u^*$  tel que  $v$  génère un mur d'épaisseur  $r$ . On peut donc définir l'intervalle  $I_v = [a_v, b_v]$  qui est non vide.

Pour tout  $t \in \mathbb{N}$ , on définit maintenant :

$$x_t = \min\{x \in \mathbb{Z}, \forall \tau \leq t, \forall y \in [\Delta_{\frac{x}{t}}(\tau) - r, \Delta_{\frac{x}{t}}(\tau) + r], \langle y, \tau \rangle \in \mathfrak{C}(v)\}$$

La position  $x_t$  est calculable pour tout  $t \in \mathbb{N}$  et donc la suite  $(\frac{x_t}{t})_t$  l'est aussi.

C'est à dire que  $\frac{x_t}{t}$  est le plus petit nombre susceptible (au temps  $t$ ) d'être une pente selon laquelle  $v$  est bloquant. Au temps  $\tau \leq t$ ,  $\frac{x_t}{t}$  est donc aussi susceptible d'être une pente selon laquelle  $v$  est bloquant donc il est plus grand que le plus petit tel nombre :  $\forall t \geq \tau, \frac{x_t}{t} \geq \frac{x_\tau}{\tau}$ . Comme  $(\frac{x_t}{t})_t$  est croissante et bornée par  $a_v$ , la suite converge vers  $\alpha \leq a_v$  et par construction  $v$  est bloquant de direction  $\Delta_\alpha$  donc  $\alpha \geq a_v$ .

Finalement  $\alpha = a_v$  et l'intervalle est fermé en  $a_v$ . Comme la suite  $(\frac{x_t}{t})_t$  est calculable et converge vers  $a_v$ ,  $a_v$  est r.e.g.. La preuve est symétrique en  $b_v$ .  $\square$

On peut maintenant prouver en utilisant ce résultat le théorème suivant :

**Théorème 3.5.2.** *Soit  $\mathcal{A}$  un automate cellulaire, il existe  $\alpha', \alpha'' \in \mathbb{R}$  tels que  $\tilde{\mathbf{A}}(\mathcal{A}) = \{\Delta_\alpha, \alpha \in |\alpha', \alpha''|\}$ . Alors  $\alpha'$  et  $\alpha''$  sont r.e. et si  $|\alpha', \alpha''|$  est fermé à gauche,  $\alpha'$  est r.e.g., si  $|\alpha', \alpha''|$  est fermé à droite,  $\alpha''$  est r.e.d..*

*Démonstration.* On prouve le résultat pour la borne gauche, la preuve est symétrique à droite. Si l'intervalle est fermé à gauche, alors il existe un mot  $u$  bloquant de direction  $\Delta_{\alpha'}$ , donc la proposition 3.5.1 dit qu'il existe  $v$  tel que  $I_v = [\alpha', b_v]$  et que  $\alpha'$  est r.e.g..

Si l'intervalle est ouvert en  $\alpha'$ , d'après la proposition 3.5.1, il existe une suite  $(v_i)_i$  de mots tels que  $I_{v_i} = [a_i, b_i]$  avec  $a_i \in ]\alpha', \alpha' + \frac{1}{i}]$ . Comme  $a_i$  est r.e.g., il existe une suite de rationnels  $(s_{i,j})_j$  qui converge vers  $a_i$ . Alors  $(s_{i,i})_i$  est une suite de rationnels qui converge vers  $\alpha'$ . Donc  $\alpha'$  est r.e.  $\square$

Dans le cas où  $\alpha' = \alpha''$ , on a donc le corollaire suivant :

**Corollaire 3.5.3.** *Soit  $\mathcal{A}$  un automate cellulaire, et  $\alpha$  un réel tel que  $\tilde{\mathbf{A}}(\mathcal{A}) = \{\Delta_\alpha\}$ , alors  $\alpha$  est calculable.*

## 3.6 Obtenir les bornes

On cherche ici à construire des automates cellulaires ayant comme ensemble de directions linéaires de points d'équicontinuité, chacun des ensembles autorisés par le théorème 3.5.2. On va montrer le résultat suivant :

**Théorème 3.6.1.** *Soient  $\alpha'$  et  $\alpha''$  des réels r.e., il existe un automate cellulaire  $\mathcal{A}$  tel que  $\tilde{\mathbf{A}}(\mathcal{A}) = \{\Delta_\alpha, \alpha \in |\alpha', \alpha''|\}$ . Si, de plus,  $\alpha'$  est r.e.g., l'intervalle peut être fermé à gauche et si  $\alpha''$  est r.e.d., il peut être fermé à droite.*

Pour prouver ce théorème, on montre la proposition suivante :

**Proposition 3.6.2.** *Pour tout réel r.e.  $0 \leq \alpha \leq 1$ , il existe un automate cellulaire  $\mathcal{A}$  tel que  $\tilde{\mathbf{A}}(\mathcal{A}) = \{\Delta_{\alpha'}, \alpha' \in |\alpha, 1|\}$ . Si  $\alpha$  est r.e.g., l'intervalle peut être fermé à gauche.*

Ensuite en augmentant le rayon et en « dilatant » l'automate cellulaire, on pourra obtenir  $\alpha$  arbitrairement grand. Une symétrie de la règle locale donnera les r.e. négatifs, puis des produits cartésiens permettront d'obtenir l'intervalle voulu.

*Démonstration.* On construit un automate pour chacun de ces ensembles. L'idée générale partage quelques principes avec celle utilisée pour la construction de l'automate  $\mathcal{A}_p$  défini pour la proposition 2.2.1 : une machinerie sur la gauche envoie des signaux qui détruisent chacun un  $\blacksquare$  d'une large zone  $\blacksquare$ . L'automate sera de rayon

1. Un état  $\boxtimes$  démarre la construction en activant une machine de Turing dont on détaillera le comportement par la suite. Une large zone d'états  $\blacksquare$  contiendra l'ensemble des conséquences de l'état  $\boxtimes$ . L'état  $\square$  est à nouveau quiescent, et constitue le fond dans lequel les signaux se déplacent. On détaille la construction pour la configuration  $c = \mathbb{N} \square \boxtimes \square \mathbb{N}$ . Les conséquences de l'état  $\boxtimes$  sont exactement l'ensemble des sites dans l'état  $\blacksquare$  pour cette configuration. On montrera ensuite que pour toute autre configuration tous ces sites sont encore dans l'état  $\blacksquare$ .

La machine de Turing va construire sur la gauche une zone de calcul qui enverra des signaux vers la zone  $\blacksquare$ . Chacun de ces signaux se déplaçant vers la droite transformera le premier  $\blacksquare$  rencontré en  $\square$ . C'est le seul cas dans lequel un état  $\blacksquare$  sera détruit. Dans tous les autres cas, lorsqu'un site est dans un état  $\blacksquare$ , il le reste, et son voisin de droite prend l'état  $\blacksquare$  quels que soient les états des autres antécédents.

On veut que la frontière de la zone  $\blacksquare$  ait pour asymptote une droite de pente  $\alpha$ . Il faut pour cela que cette zone reçoive un nombre de signaux qui tend vers  $\alpha T$  pour  $T$  étapes.

### Densité de signaux à envoyer

Comme la frontière doit tendre vers une droite de pente  $\alpha$ , pour  $N$  signaux traversant une cellule donnée, elle ne recevra que  $(1 - \alpha)N$  signaux. Il faut donc envoyer  $\beta = \frac{\alpha}{1-\alpha}$  signaux par unité de temps. Puisque  $\alpha$  est r.e.,  $\beta$  l'est aussi. Donc il existe une suite calculable  $\beta_i$  de rationnels qui tend vers  $\beta$ . Ces  $\beta_i$  seront les densités successives de signaux que l'on va envoyer.

La figure 3.5 illustre ce principe.

### Envoi des signaux

La construction est illustrée par la figure 3.6. La machine construit successivement sur la gauche des colonnes de taille  $2^i$  pour  $i \geq 0$ , dont la première cellule contient l'état  $\uparrow$ . Dans chacune de ces colonnes, un signal (en pointillé sur la figure 3.6) rebondit entre les deux bords à vitesse 1. Le signal fait donc un aller-retour dans la colonne  $i$  en  $2^{i+1}$  étapes.

La machine doit synchroniser toutes les colonnes, donc le signal de la colonne  $i + 1$  doit être sur la première cellule de sa colonne seulement lors d'étapes où celui de la colonne  $i$  est aussi sur sa première cellule. Comme les temps d'aller-retour sont multipliés par 2 à chaque nouvelle colonne, Lorsqu'un signal est sur la première cellule de sa colonne, il en est de même pour toutes les colonnes précédentes.

Pour tout  $i \in \mathbb{N}$ , après avoir construit la colonne  $i$ , la machine calcule une approximation de  $\beta$ , c'est à dire qu'elle calcule les  $i$  premiers bits de  $\beta_i$ . Puis elle « active »

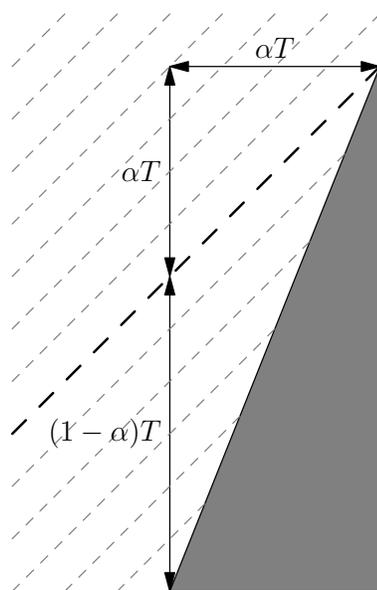


FIGURE 3.5 – Des signaux (en pointillé) viennent de la gauche et creusent la zone  $\blacksquare$ . Seuls les  $(1 - \alpha)T$  premiers signaux rencontrent effectivement la zone  $\blacksquare$  en temps  $T$ .

les colonnes  $k \leq i$  telles que le  $k$ -ième bit de  $\beta_i$  est égal à 1. Activer une colonne revient à changer l'état  $\uparrow$  de sa première cellule en  $\blacksquare$ .

Pour chaque colonne activée, à chaque fois que le signal interne de la colonne est sur la première cellule, un signal (en pointillé gris sur la figure) à vitesse 1 est envoyé vers la droite qui remplacera le premier état  $\blacksquare$  rencontré par un  $\square$ . Donc la colonne  $i$  activée envoie un signal toutes les  $2^{i+1}$  étapes.

Ainsi la densité de signaux produits vaut  $\beta_i$  à  $2^{-(i+1)}$  près. Il reste à s'assurer que les signaux ne se superposent pas. La synchronisation des colonnes nous assure cette propriété. En effet, si la colonne  $i$  est activée, elle envoie des signaux qui passeront la colonne 0 aux temps  $(n \times 2^{i+1} + 2^i - 1)$  pour tous  $n \in \mathbb{N}$ . En particulier, pour générer une densité 1 de signaux, il faudrait que toutes les colonnes soient créées et qu'elles soient toutes actives.

Comme  $(\beta_i)_i$  tend vers  $\beta$ , pour tout  $k \in \mathbb{N}$ , il existe  $i_k$  minimal tel que  $\forall i \geq i_k$ ,  $\beta_{i_k}$  et  $\beta$  coïncident sur les  $k$  premiers bits. Ainsi, les  $k$  premières colonnes sont définitivement fixées après un certain temps, et la densité de signaux envoyés converge vers  $\beta$ .

### $\tilde{\mathbf{A}}(\mathcal{A})$

La zone de calcul construite par la machine n'est jamais influencée par les états situés sur sa gauche sauf par l'état  $\blacksquare$ . Si cet état apparaît à gauche, il détruit toute la zone de construction. Un cas peut alors poser problème : si la zone de construction est détruite par un  $\blacksquare$  et que de nombreux signaux arrivent de la gauche, la pente dessinée par la frontière de la zone  $\blacksquare$  peut être trop importante (par exemple 1 si des signaux arrivent à chaque étape). Pour éviter cela, on protège toute la construction dans des compteurs créés par  $\boxtimes$ , ainsi, seule une construction similaire sur la gauche pourra interférer, et dans ce cas, la pente produite à gauche ne sera pas trop forte.

Puisque la frontière gauche de ses conséquences a pour asymptote une droite de pente  $\alpha$ ,  $\boxtimes$  est un mot bloquant pour toutes les droites de pente comprise entre  $\alpha$  et 1. En revanche la borne  $\alpha$  n'est pas nécessairement fermée. Elle l'est lorsque  $\alpha$  est r.e.g. si on utilise une suite  $(\alpha_i)_i$  croissante (donc  $(\beta_i)_i$  aussi). La frontière des conséquences aura donc toujours une pente inférieure à  $\alpha$ .

### Borne droite

Pour la borne droite de l'intervalle, il faut faire une construction similaire de l'autre côté de la zone  $\blacksquare$ . Pour l'intervalle  $[0, \alpha]$  avec  $0 < \alpha < 1$ , les signaux se déplacent vers la gauche et ajoutent des états  $\blacksquare$  au lieu de les détruire. On a alors besoin d'un rayon plus grand, et de construire des colonnes « diagonales ». Il est nécessaire aussi de protéger la construction entre des compteurs comme vu en 2.3.  $\square$

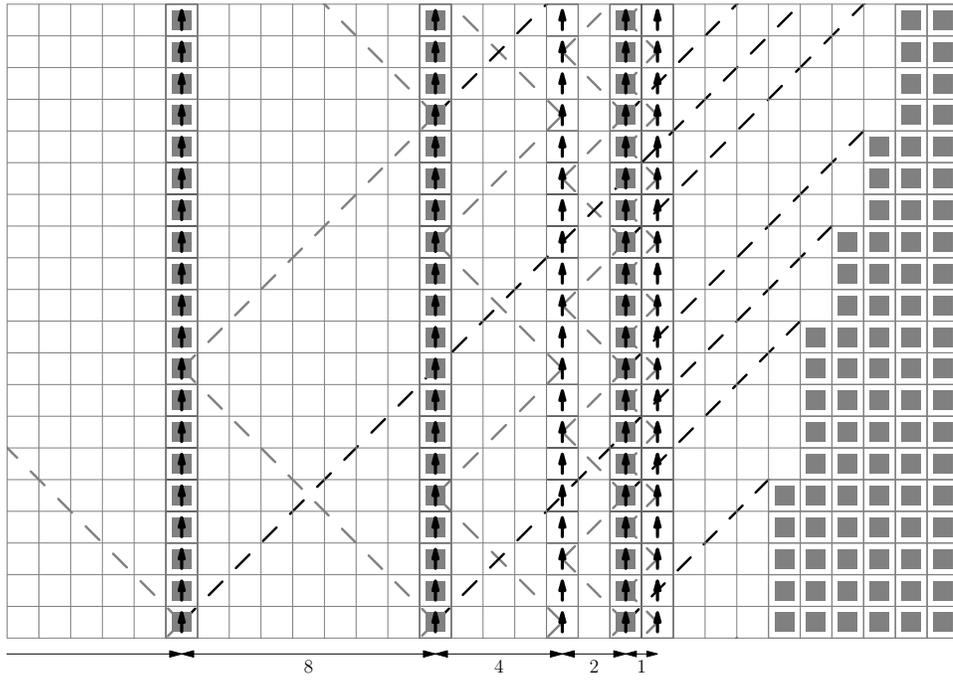


FIGURE 3.6 – Les premières colonnes sont formées et les colonnes 1, 3 et 4, de tailles 2, 8 et 16 sont activées. Si aucune autre colonne n'est activée, la densité de signaux envoyés sera donc  $\frac{1}{2} \left( \frac{1}{2^1} + \frac{1}{2^3} + \frac{1}{2^4} \right)$ .



## Chapitre 4

# Ensembles $\mu$ -limite

On revient dans ce chapitre sur la remarque 3.4.1 qui affirme que lorsqu'un automate cellulaire a une direction bloquante qui n'est pas une droite de pente rationnelle, les images successives de toutes les configurations contenant des murs arbitrairement loin à droite et à gauche, tendent à être uniformes. Cette propriété est en particulier vérifiée par toutes les configurations choisies aléatoirement avec une probabilité uniforme sur l'ensemble des états. Si on considère la mesure uniforme associée à cette distribution de probabilité, l'ensemble des configurations qui ne vérifient pas cette propriété est de mesure nulle.

On peut donc parler de comportement typique pour l'automate cellulaire qui, s'il n'est pas nilpotent, présente une forme affaiblie de nilpotence : une « presque-nilpotence ». On peut généraliser cette idée en s'intéressant à l'ensemble des mots qui apparaissent arbitrairement tard et souvent dans l'ensemble des diagrammes espace-temps d'un automate cellulaire. On peut de cette manière définir un langage et donc un sous-shift associé que l'on va ici étudier. On verra que ce sous-shift est différent de l'ensemble limite mais le contient. En partant du cas des automates cellulaires ayant un mot bloquant, on s'intéressera à ses propriétés, ainsi qu'aux sous-shifts qui peuvent être construits de cette manière.

Les ensembles  $\mu$ -limite ont été introduits par P. Kurka et A. Maass [23] comme un raffinement des  $\mu$ -attracteurs de M. Hurley [19].

Les résultats de ce chapitre ont été publiés dans [3] et [9].

### 4.1 Définitions

Il existe différentes définitions d'ensembles de configurations obtenus après un temps infini pour un automate cellulaire. La définition classique est celle d'ensemble limite qui est le sous-shift dont le langage est l'ensemble des mots qui peuvent

apparaître arbitrairement tard dans un diagramme espace-temps.

P. Kůrka et A. Maass introduisent une nouvelle notion en 2000 [23], qui correspond aux images asymptotiques de configurations aléatoires. L'ensemble  $\mu$ -limite est un sous-shift que l'on définit à l'aide d'un langage de mots interdits. Un mot sera interdit s'il est *évanescent*, c'est à dire si sa probabilité d'apparaître dans les images d'une configuration tend vers 0 au cours du temps. Formellement, il faut considérer une mesure, et on ne s'intéressera dans tout ce qui suit qu'à la mesure uniforme  $\mu$  sur  $X^{\mathbb{Z}}$ . Pour tout mot  $u \in X^*$ , on a donc  $\mu([u]) = (\frac{1}{|X|})^{|u|}$ .

On note, pour tout mot  $u \in X^*$ ,  $\text{Ant}^t(\mathcal{A}, u) = \{v \in X^{|u|+2rt}, \forall c \in [v], \mathcal{A}^t(c) \in [u]\}$  l'ensemble des mots antécédents de  $u$  à l'étape  $t$  pour l'automate cellulaire  $\mathcal{A}$ . On se contentera de la notation  $\text{Ant}^t(u)$  lorsqu'il n'y aura pas de confusion possible. Pour un mot  $u$ , la quantité qui permettra de décider si le mot apparaît dans l'ensemble  $\mu$ -limite est le nombre de ses antécédents à chaque étape. Pour  $t \in \mathbb{N}$ , on note

$$\mathcal{A}^t \mu([u]) = \mu(\mathcal{A}^{-t}([u])) = \mu(\{c \in X^{\mathbb{Z}}, \mathcal{A}^t(c) \in [u]\}) = \frac{|\text{Ant}^t(u)|}{|X|^{|u|+2rt}}$$

**Définition 4.1.1.** Soit  $\mathcal{A}$  un automate cellulaire.

- Un mot  $u \in X^*$  est dit évanescent lorsque  $\lim_{t \rightarrow \infty} \mathcal{A}^t \mu([u]) = 0$ .
- On définit le langage persistant comme l'ensemble des mots non évanescents :

$$L_{\mu}(\mathcal{A}) = \{u \in X^*, \mathcal{A}^t \mu([u]) \not\rightarrow 0\}$$

- L'ensemble  $\mu$ -limite est alors le sous-shift associé au langage persistant

$$\Lambda_{\mu}(\mathcal{A}) = \{c \in X^{\mathbb{Z}}, \forall i \leq j \in \mathbb{N}, c_{[i,j]} \in L_{\mu}(\mathcal{A})\}$$

$\Lambda_{\mu}(\mathcal{A})$  est défini comme un sous-shift à l'aide d'un langage de mots interdits, il est donc naturellement invariant par décalage et fermé (et même compact). Enfin, il est stable par  $\mathcal{A}$  puisque l'image d'une configuration sans mot évanescent ne contient pas non plus de mot évanescent.

La proposition suivante compare ces deux ensembles :

**Proposition 4.1.1.** *Pour un automate cellulaire  $\mathcal{A}$ ,  $\Lambda_{\mu}(\mathcal{A}) \subseteq \Lambda(\mathcal{A})$ .*

*Démonstration.* On prend  $u \notin \mathcal{L}(\Lambda(\mathcal{A}))$ . Il existe  $t_0 \in \mathbb{N}$  tel que  $\forall t \geq t_0, \forall c \in X^{\mathbb{Z}}, \mathcal{A}^t(c) \notin [u]$ . Donc nécessairement  $\mathcal{A}^t \mu([u]) \rightarrow_{t \rightarrow \infty} 0$  et  $u \notin L_{\mu}(\mathcal{A})$ .  $\square$

De plus, ces ensembles ne sont jamais vides, on démontre ce résultat pour l'ensemble  $\mu$ -limite.

**Proposition 4.1.2.** *Pour tout automate cellulaire  $\mathcal{A}$ , il existe une configuration dans l'ensemble  $\mu$ -limite de  $\mathcal{A}$ .*

*Démonstration.* Clairement, le mot vide appartient à  $L_\mu(\mathcal{A})$ , on montre alors que pour tout mot  $u \in L_\mu(\mathcal{A})$ , il existe  $s_1, s_2 \in X$  tels que  $s_1us_2 \in L_\mu(\mathcal{A})$ . Supposons le contraire :  $\forall s_1, s_2 \in X, \mathcal{A}^n\mu([s_1us_2]) \rightarrow_n 0$ . On a alors :

$$\sum_{s_1, s_2 \in X} \mu(\mathcal{A}^{-n}([s_1us_2])) = \mu\left(\bigcup_{s_1, s_2 \in X} \mathcal{A}^{-n}([s_1us_2])\right) = \mu(\mathcal{A}^{-n}([u]))$$

Il s'ensuit que  $\mathcal{A}^n\mu([u]) \rightarrow_n 0$ , et donc  $u \notin L_\mu(\mathcal{A})$ .

En partant, du mot vide, on peut donc construire récursivement une configuration dont tous les facteurs sont dans  $L_\mu(\mathcal{A})$ . □

### 4.1.1 Exemples

On va maintenant regarder deux exemples d'automates cellulaires, et les ensembles associés à ces exemples. Dans un premier temps, on s'intéresse à l'automate MAX :

*Exemple 4.1.1.* On a vu en 1.2.4 que l'ensemble limite de MAX est :

$$\Lambda(\text{MAX}) = (\mathbb{N}_{\blacksquare} \blacksquare^* \blacksquare^{\mathbb{N}}) \cup (\mathbb{Z}) \cup (\mathbb{N}_{\blacksquare} \blacksquare^{\mathbb{N}}) \cup (\mathbb{N}_{\blacksquare} \blacksquare^{\mathbb{N}})$$

Pour l'ensemble  $\mu$ -limite, clairement  $\forall t \in \mathbb{N}, \text{Ant}^t(\blacksquare) = \{\blacksquare^{2t+1}\}$ , et donc

$$\mathcal{A}^t\mu([u]) = \frac{1}{2^{2t+1}} \rightarrow_{t \rightarrow \infty} 0$$

Ce qui signifie que  $\blacksquare \notin L_\mu(\mathcal{A})$ . Comme l'ensemble  $\mu$ -limite n'est pas vide d'après la proposition 4.1.2, l'unique configuration ne contenant pas de  $\blacksquare$  appartient à  $\Lambda_\mu(\mathcal{A})$ . Finalement  $\Lambda_\mu(\mathcal{A}) = \{\blacksquare^{\mathbb{Z}}\}$ .

*Exemple 4.1.2.* On regarde un deuxième exemple classique, l'automate 184 de la classification de Wolfram [41], qui est considéré comme un modèle simple de simulation du trafic routier. Il n'y a toujours que deux états :  $X = \{\square, \blacksquare\}$ . La règle est la suivante :



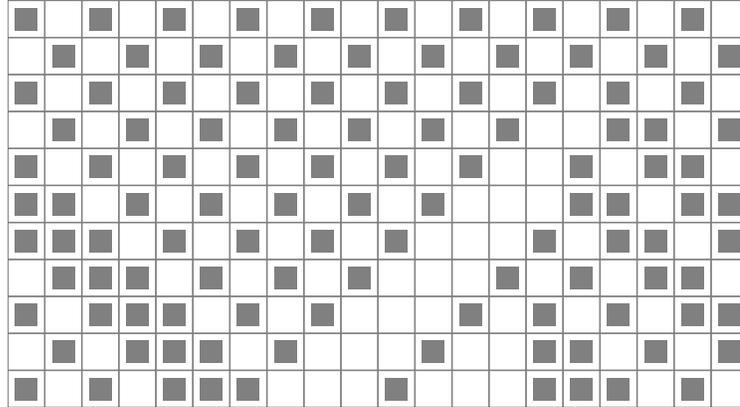


FIGURE 4.1 – Un extrait de diagramme espace-temps de l'automate 184.

On peut voir les états  $\blacksquare$  comme des voitures qui avancent lorsqu'elles voient un espace vide ( $\square$ ) devant elles, et qui attendent sinon. Ainsi, on peut faire correspondre à chaque état  $\blacksquare$  d'une configuration, un nouvel état  $\blacksquare$  dans l'image par l'automate. Soit à la même cellule, si la voiture n'a pas avancé, soit à la cellule suivante. Pour l'image d'un mot fini (de longueur 2 de moins), il y a au plus une voiture qui est sortie du mot en avançant à la fin du mot et une qui n'a pas pu avancer au début du mot. Donc le nombre d'états  $\blacksquare$  a diminué de 2 au plus en une étape. Le fonctionnement de l'automate est illustré par la figure 4.1.

On veut maintenant montrer que le mot  $u = \square\square$  n'est pas persistant. En étudiant tous les cas, on constate que pour cet automate, si deux sites consécutifs  $\langle x, t \rangle$  et  $\langle x+1, t \rangle$  ont pour état  $\square$ , alors à l'étape précédente, les sites  $\langle x-1, t-1 \rangle$  et  $\langle x, t-1 \rangle$  ont aussi cet état, c'est à dire que tout mot de  $X^4$  d'image  $u$  commence par  $u$ . De même pour ses antécédents à l'étape  $t$  :  $\text{Ant}^t(u) \subseteq uX^{2t}$ .

Soit  $v \in X^{2t+1}$  tel que  $\square v \in \text{Ant}^t(u)$ . On va maintenant montrer que tous les préfixes de  $v$  contiennent plus de  $\square$  que de  $\blacksquare$ . On suppose le contraire et on prend  $k \in \mathbb{N}$ , le plus petit entier tel que le préfixe de  $v$  de longueur  $k$ ,  $w = v_{[0, k-1]}$ , contienne strictement plus de  $\blacksquare$  que de  $\square$ . Comme  $k$  est le plus petit tel entier,  $w_{k-1} = w_k = \blacksquare$ . Si  $k \geq 3$ , on peut aussi supposer que  $w_{k-3} \neq \square$  ou  $w_{k-2} \neq \square$  toujours parce que  $k$  est minimal. Ainsi si  $w' \in X^{k-1}$  est l'image de  $w$  par  $\mathcal{A}$ , alors  $w'_{k-2} = w'_{k-3} = \blacksquare$  :



Tous les antécédents finissent donc par deux états  $\blacksquare$ . Entre  $w$  et  $w'$ , un seul état  $\blacksquare$  a disparu (puisque le mot commence par  $u$ , aucun état  $\blacksquare$  n'a été perdu au début du mot), on peut donc affirmer que  $w'$  contient lui aussi plus de  $\blacksquare$  que de  $\square$ . De plus,  $w'$  est le plus petit mot de  $\text{Ant}^{t-1}(u)$  ayant cette propriété, sinon  $w$  ne serait pas minimal.

Par récurrence on peut se ramener à  $k = 0$  ou  $k = 1$ , on montre donc qu'un antécédent de  $u$  commence par  $\square\blacksquare$  ou par  $\square\blacksquare\blacksquare$ , ce qui est impossible dans les deux cas.

Donc les antécédents de  $u$  sont formés d'un état  $\square$  suivi d'un mot dont aucun préfixe ne contient plus de  $\blacksquare$  que de  $\square$ . Ces mots sont exactement les préfixes des mots bien parenthésés où  $\square$  et  $\blacksquare$  sont respectivement les parenthèses ouvrante et fermante. La mesure de cet ensemble de mots parmi les mots d'une taille donnée tend vers 0 [12], donc  $\mathcal{A}^n\mu(\llbracket\square\square\rrbracket) \rightarrow_n 0$ .

L'automate présente une symétrie : en permutant les états, on obtient l'automate cellulaire inversé : les voitures se déplacent vers la gauche. Cette symétrie permet d'affirmer que  $\mathcal{A}^n\mu(\llbracket\blacksquare\blacksquare\rrbracket) \rightarrow_n 0$  aussi. Comme l'ensemble  $\mu$ -limite n'est pas vide, il contient une configurations sans  $\square\square$  ni  $\blacksquare\blacksquare$ , c'est-à-dire  $(\square\blacksquare)^{\mathbb{Z}}$  centré en la cellule 0 ou la cellule 1. L'invariance par décalage implique que l'ensemble  $\mu$ -limite contient exactement les deux configurations  $\{(\square\blacksquare)^{\mathbb{Z}}, (\blacksquare\square)^{\mathbb{Z}}\}$ .

L'ensemble limite de cet automate est plus grand, il contient par exemple les deux configurations uniformes et des configurations non périodiques. Il est donc en particulier infini (puisque'il contient des configurations non périodiques comme  ${}^{\mathbb{N}}\square\blacksquare\square^{\mathbb{N}}$ ).

### 4.1.2 Densités

On va voir ici une autre manière de définir les ensembles  $\mu$ -limite d'automates cellulaires. On a défini  $\Lambda_\mu$  en interdisant les mots n'ayant que peu d'antécédents, cela revient à considérer les images de mots finis ayant une forte probabilité d'apparaître. On va montrer qu'il existe des configurations « témoins » dont les images ne contiennent avec une fréquence importante que les mots persistants.

On définit une notion de densité sur les mots et les configurations comme suit :

**Définition 4.1.2.** Pour tous mots  $u, v \in X^*$ .

- On note

$$|v|_u = |\{i, 0 \leq i \leq |v| - |u| \text{ et } v_{[i, i+|u|-1]} = u\}|$$

le nombre d'occurrences de  $u$  dans  $v$ .

- On note

$$d_v(u) = \frac{|v|_u}{|v| - |u| + 1}$$

la densité de ce mot  $u$  dans le mot  $v$ .

- Pour une configuration  $c \in X^{\mathbb{Z}}$ , la densité de  $u$  dans  $c$  est :

$$d_c(u) = \limsup_{n \rightarrow +\infty} d_{c_{[-n,n]}}(u) = \limsup_{n \rightarrow +\infty} \frac{|c_{[-n,n]}|_u}{2n + 2 - |u|}$$

- Pour une demi-configuration  $c \in X^{\mathbb{N}}$ , la densité de  $u$  dans  $c$  est :

$$d_c(u) = \limsup_{n \rightarrow +\infty} d_{c_{[0,n]}}(u) = \limsup_{n \rightarrow +\infty} \frac{|c_{[0,n]}|_u}{n + 2 - |u|}$$

Si on considère un ensemble préfixe fini  $W \subset X^*$ , on note  $S = \max_{u \in W} |u|$ .

- Pour un mot  $v \in X^*$ , on note  $|v|_W = \sum_{u \in W} |v|_u$  le nombre d'occurrence de mots de  $W$  dans  $v$ .
- Toujours avec  $v \in X^*$ , on note  $d_v(W) = \sum_{u \in W} d_v(u)$  la densité de l'ensemble  $W$  dans le mot  $v$ .
- Pour une configuration  $c \in X^{\mathbb{Z}}$ , la densité de  $W$  dans  $c$  est :

$$d_c(W) = \limsup_{n \rightarrow +\infty} d_{c_{[-n,n]}}(W) = \sum_{u \in W} d_c(u)$$

- Pour une demi-configuration  $c \in X^{\mathbb{N}}$ , la densité de  $W$  dans  $c$  est :

$$d_c(W) = \limsup_{n \rightarrow +\infty} d_{c_{[0,n]}}(W) = \sum_{u \in W} d_c(u)$$

Ces limites supérieures existent toutes puisqu'elles concernent des suites bornées.

Certaines configurations contiennent tous les mots avec une grande densité, ce sont ces configurations que l'on utilisera :

**Définition 4.1.3.** Une configuration  $c$  est dite *générique* lorsqu'il existe  $b \geq 1$  tel que

$$\forall l \in \mathbb{N}, \forall u \in X^l, \frac{1}{b|X|^l} \leq d_c(u) \leq \frac{b}{|X|^l}$$

Si une configuration  $c$  est telle que tout mot  $u \in X^l$  a pour densité  $\frac{1}{|X|^l}$  pour toute longueur  $l \in \mathbb{N}$  ( $b = 1$  dans la définition de configuration générique),  $c$  est dite *normale*.

Les configurations normales sont donc génériques.

Ces configurations sont intéressantes, car en contenant tous les mots de même taille avec une densité proche, leurs images vont contenir avec une grande densité tous les mots ayant de nombreux antécédents.

**Lemme 4.1.3.** *Soit  $\mathcal{A}$  un automate cellulaire,  $u$  un mot fini sur  $X$  et  $c$  une configuration générique, alors :*

$$u \in L_\mu(\mathcal{A}) \Leftrightarrow d_{\mathcal{A}^n(c)}(u) \xrightarrow{n \rightarrow \infty} 0$$

*Démonstration.* Soit  $n \in \mathbb{N}$ , on va montrer que

$$\frac{|\text{Ant}^n(u)|}{b|X|^{|u|+2rn}} \leq d_{\mathcal{A}^n(c)}(u) \leq \frac{b|\text{Ant}^n(u)|}{|X|^{|u|+2rn}}$$

On a :

$$d_{\mathcal{A}^n(c)}(u) = d_c(\text{Ant}^n(u)) = \sum_{v \in \text{Ant}^n(u)} d_c(v)$$

Or  $c$  est générique donc il existe  $b > 0$  tel que pour tout  $v \in \text{Ant}^n(u)$ ,

$$\frac{1}{b|X|^{|v|}} < d_c(v) \leq \frac{b}{|X|^{|v|}}$$

C'est à dire

$$\sum_{v \in \text{Ant}^n(u)} \frac{1}{b|X|^{|v|}} \leq d_{\mathcal{A}^n(c)}(u) \leq \sum_{v \in \text{Ant}^n(u)} \frac{b}{|X|^{|v|}}$$

Et donc

$$\frac{|\text{Ant}^n(u)|}{b|X|^{|u|+2rn}} \leq d_{\mathcal{A}^n(c)}(u) \leq \frac{b|\text{Ant}^n(u)|}{|X|^{|u|+2rn}}$$

Ainsi on a l'équivalence

$$\limsup_{n \rightarrow +\infty} d_{\mathcal{A}^n(c)}(u) = 0 \Leftrightarrow \limsup_{n \rightarrow +\infty} \frac{|\text{Ant}^n(u)|}{|X|^{|u|+2rn}} = 0$$

□

Les configurations génériques, et parmi elles en particulier les configurations normales, sont donc les témoins de l'ensemble  $\mu$ -limite. Comme les configurations normales sont aussi les configurations obtenues par tirage aléatoire selon  $\mu$ , l'ensemble  $\mu$ -limite correspond bien, intuitivement, à l'ensemble des comportements typiques d'un automate cellulaire.

### 4.1.3 $\mu$ -nilpotence

La nilpotence d'un automate cellulaire est définie comme le fait de n'avoir qu'une seule configuration dans son ensemble limite, la notion de nilpotence approchée que l'on veut établir à l'aide de l'ensemble  $\mu$ -limite peut alors s'énoncer comme suit :

**Définition 4.1.4.** Un automate cellulaire  $\mathcal{A}$  est dit  $\mu$ -nilpotent lorsque  $\Lambda_\mu(\mathcal{A})$  est réduit à une seule configuration.

L'invariance par décalage de  $\Lambda_\mu(\mathcal{A})$  implique que cette unique configuration est égale à tous ses décalés et donc qu'elle est uniforme.

Toutefois un automate cellulaire est nilpotent dès lors que son ensemble limite est fini, donc les ensembles limite finis ne contiennent en réalité qu'une configuration uniforme [7]. Cette propriété n'est pas vérifiée pour les ensembles  $\mu$ -limite, par exemple l'automate 184 de la classification de Wolfram a, comme on l'a vu, un ensemble  $\mu$ -limite réduit à deux configurations non uniformes. Un ensemble  $\mu$ -limite peut donc être fini sans que l'automate cellulaire soit  $\mu$ -nilpotent, il peut aussi ne contenir aucune configuration uniforme.

Plusieurs définitions pourraient donc être acceptables pour la  $\mu$ -nilpotence, entre autres :

- l'ensemble  $\mu$ -limite est fini ;
- l'ensemble  $\mu$ -limite ne contient qu'une configuration périodique et ses décalés ;
- l'ensemble  $\mu$ -limite ne contient que des configurations uniformes.

Pour la dernière de ces propositions, on peut donner un exemple d'automate cellulaire dont l'ensemble  $\mu$ -limite contient exactement deux configurations uniformes. Pour cela, on reprend la construction des compteurs définis en 2.3, et à l'intérieur des compteurs, on ne met que des états ■, qui à l'étape suivante deviendront ■ et qui alterneront ainsi. La figure 4.2 illustre le fonctionnement de cet automate. D'après le lemme 4.1.3, le langage  $\mu$ -limite correspond aux mots qui apparaissent avec une forte densité dans les images des configurations normales. Or, ici, les configurations normales contiennent des états de débuts de compteurs arbitrairement loin sur la gauche et sur la droite. Donc, toute cellule finit par être à l'intérieur de compteurs, et comme ceux-ci sont synchronisés puisqu'ils ont démarré avec la configuration initiale, les cellules entre compteurs sont alternativement toutes  $R$  ou toutes  $B$ .

Un résultat majeur concernant la  $\mu$ -nilpotence a été prouvé par L. Boyer, V. Poupet et G. Theyssier [4] :

**Théorème 4.1.4.** *L'ensemble des automates cellulaires  $\mu$ -nilpotents n'est pas récursif.*

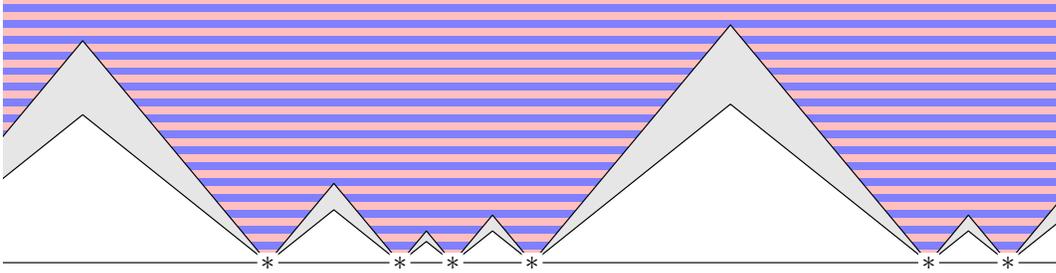


FIGURE 4.2 – Automate cellulaire dont l'ensemble  $\mu$ -limite ne contient que deux configurations uniformes.

## 4.2 Mots bloquants

Dans cette partie, on s'intéresse à des automates cellulaires ayant un mot bloquant selon une direction  $h \in \mathcal{F}$ . Il est montré [4] que dans le cas où  $h$  est égale à  $\Delta_0$ , le langage  $\mu$ -limite de l'automate est exactement l'ensemble des mots pouvant apparaître arbitrairement tard entre deux mots bloquants. Le résultat et la preuve sont directement adaptables à toutes les directions.

Ce résultat nécessite un lemme qui affirme que si un mot apparaît dans le langage  $\mu$ -limite, alors il a des antécédents arbitrairement loin contenant deux occurrences de n'importe quel mot à distance finie :

**Lemme 4.2.1.** *Soit  $\mathcal{A}$  un automate cellulaire,  $h \in \mathcal{F}$ , et  $v \in L_\mu(\mathcal{A})$ , alors pour tout mot  $u \in X^*$ , il existe  $K \geq |v| + |u|$ ,  $x \in [-\frac{K}{2}, \frac{K}{2}]$ ,  $w \in X^{K-|u|}$  et une suite  $(t_n)_n$  tels que :*

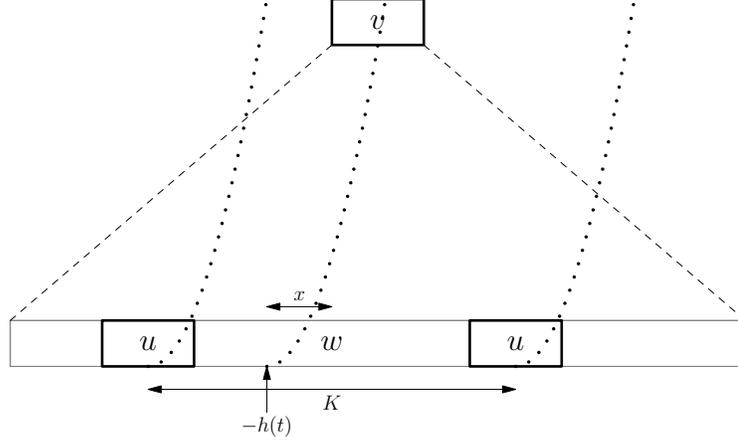
$$\mathcal{A}^{-t_n}([v]) \cap [uwu]_{x-h(t_n)} \neq \emptyset$$

La figure 4.3 illustre la propriété du lemme.

*Démonstration.* Supposons que cela soit faux :  $\forall K \geq |v| + |u|, \forall x \in [-\frac{K}{2}, \frac{K}{2}], w \in X^{K-|u|}, \exists t_0 \in \mathbb{N}$  tels que :

$$\forall t \geq t_0, \mathcal{A}^{-t}([v]) \cap [uwu]_{x-h(t)} = \emptyset$$

Cette propriété affirme que dans tout antécédent de  $v$ , il existe un large mot qui ne contient pas de  $u$  dans sa première moitié ou dans sa seconde moitié. Dans ce cas, on va montrer que les préimages de  $[v]$  sont incluses dans des suites d'ensembles dont la mesure tend vers 0. On considère des ensembles particuliers de configurations qui ne contiennent pas d'occurrences de  $u$  en de nombreux endroits. Pour  $t$  assez grand,

FIGURE 4.3 – Un antécédent de  $v$  au temps  $t$ .

l'ensemble des préimages de  $[v]$  est inclus dans ces ensembles.

Pour tout  $j \in \mathbb{N}$ , il existe  $t_j \in \mathbb{N}$  tel que  $\forall t \geq t_j$  on ait :

$$\mathcal{A}^{-t}([v]) \subseteq \Sigma_j = \left[ (X^{|u|} \setminus \{u\})^j X^{|v|+j|u|} \right]_{-h(t)} \cup \left[ X^{|v|+j|u|} (X^{|u|} \setminus \{u\})^j \right]_{-h(t)}$$

Or la mesure de  $\Sigma_j$  est égale à  $2(1 - \mu(\{u\}))^j$ . Finalement,  $\mathcal{A}^t \mu([v]) \rightarrow 0$ .  $\square$

**Proposition 4.2.2.** *Pour tout automate cellulaire  $\mathcal{A}$  tel que  $u \in X^*$  est un mot bloquant selon la direction  $h \in \mathcal{F}$ , le langage  $L_\mu(\mathcal{A})$  est l'ensemble des mots  $v \in X^*$  tels que  $\exists w \in X^*, \forall t \in \mathbb{N}, \exists t_0 \geq t, \forall c \in [uwu]$  :*

$$\exists x \in \left[ h(t_0) - \frac{|u| + |w|}{2} + \lfloor \frac{|v|}{2} \rfloor, h(t_0) + \frac{|u| + |w|}{2} - \lceil \frac{|v|}{2} \rceil \right], \sigma^x(\mathcal{A}^{t_0}(c)) \in [v]$$

*Démonstration.* La figure 4.4 illustre la propriété, le langage  $\mu$ -limite est constitué des mots apparaissant infiniment souvent entre deux murs. On commence par montrer que tous ces mots appartiennent à  $L_\mu(\mathcal{A})$ . Soit  $v \in X^*$  un tel mot, il existe une suite  $(s_n)_n$  et un mot  $w$  tels que  $v$  apparaît entre les deux murs générés par  $uwu$  à l'étape  $s_n$ . Ainsi pour toute configuration normale  $c \in X^{\mathbb{Z}}$ ,  $d_{\mathcal{A}^{s_n}(c)}(v) \geq d_c(uwu) > 0$ .

Soit maintenant un mot  $v \in L_\mu(\mathcal{A})$ . On applique le lemme précédent avec le mot bloquant  $u$  : il existe  $K \geq |v| + |u|$ ,  $x \in [-\frac{K}{2}, \frac{K}{2}]$ ,  $w \in X^{K-|u|}$  et une suite  $(t_n)_n$  tels

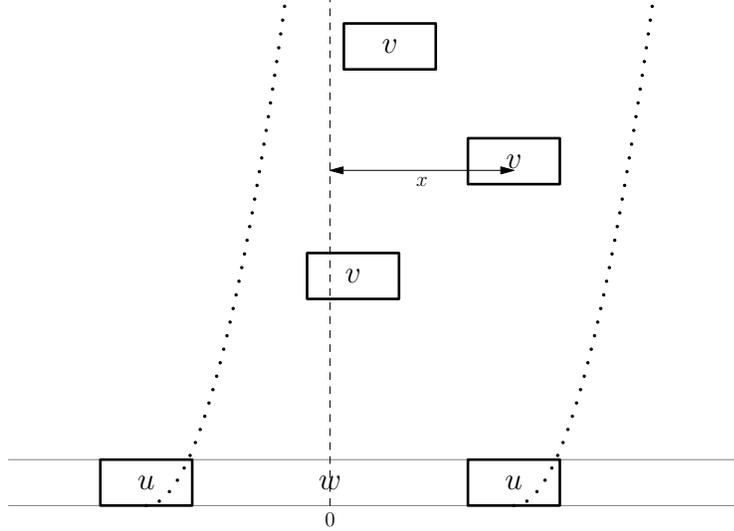


FIGURE 4.4 – Le mot  $v$  apparaît une infinité de fois entre les deux murs, donc il appartient à  $L_\mu(\mathcal{A})$ .

que :

$$\forall n \in \mathbb{N}, \mathcal{A}^{-t_n}([v]) \cap ([uvw]_{x-h(t_n)}) \neq \emptyset$$

Soit  $t \in \mathbb{N}$ , il existe  $m \in \mathbb{N}$  tel que  $t_m \geq t$ , et on prend

$$c \in \mathcal{A}^{-t_m}([v]) \cap ([uvw]_{x-h(t_m)})$$

Les deux occurrences de  $u$  engendrent des murs de directions  $h + x - h(t_m) - \frac{K}{2}$  et  $h + x - h(t_m) + \frac{K}{2}$ . En particulier en  $t_m$ , les murs sont aux abscisses  $x - \frac{K}{2} \leq 0$  et  $x + \frac{K}{2} \geq |v|$ .

On prend alors  $c' \in [uvw]$ . Comme  $u$  est un mot bloquant selon la direction  $h$ , on a

$$\left( \sigma^{x-h(t_m)}(\mathcal{A}^{t_m}(c')) \right)_{[-\lfloor \frac{|v|}{2} \rfloor, \lceil \frac{|v|}{2} \rceil]} = \mathcal{A}^{t_m}(c)_{[-\lfloor \frac{|v|}{2} \rfloor, \lceil \frac{|v|}{2} \rceil]} = v$$

□

*Remarque 4.2.1.* Si un automate cellulaire a un mot bloquant  $u$  selon une pente  $h$ , l'ensemble des états des sites des conséquences de tout mot  $uvw, v \in X^*$  est ultimement périodique.

Il l'est si  $h$  est équivalente à une droite de pente rationnelle parce que l'ensemble  $\mathfrak{C}(uvw)$  contient tous les sites entre deux droites de même pente rationnelle (donc de

représentation périodique). Ainsi  $\mathfrak{C}(uvw)$  contient un ensemble ultimement périodique de sites dont les états sont aussi ultimement périodiques (on le voit en regardant la configuration  $(uv)^{\mathbb{Z}}$  par exemple).

Si  $h$  n'est pas équivalente à une droite de pente rationnelle, la proposition 3.4.1 permet d'obtenir la même conclusion, puisque les conséquences sont uniformes par ligne.

Pour tout tel mot  $uvw$ , il existe donc une période spatio-temporelle dans les conséquences, et la proposition précédente affirme que tous les mots de cette période sont dans l'ensemble  $\mu$ -limite.

On a donc le corollaire suivant :

**Corollaire 4.2.3.** *Soit  $\mathcal{A}$  un automate cellulaire ayant un mot bloquant selon une direction  $h$ . Le langage  $\mu$ -limite de  $\mathcal{A}$  est récursivement énumérable.*

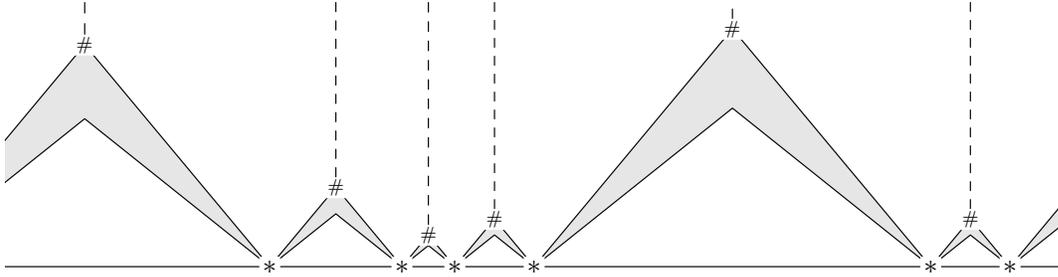
*Démonstration.* Pour obtenir ce résultat, en connaissant l'automate  $\mathcal{A}$  et un mot bloquant  $u$  sur  $\mathcal{A}$ , pour chaque mot  $w \in X^*$ , on calcule une période des conséquences de  $uvw$  situées entre les deux murs (atteinte en au plus  $|X|^{|w|+|u|}$  étapes). On énumère alors tous les mots de cette période.  $\square$

*Remarque 4.2.2.* Dans le cas où  $h$  n'est pas équivalente à une droite de pente rationnelle, ces mots sont nécessairement uniformes, mais comme vu dans la partie 4.2, il peut y en avoir plusieurs. Avec la définition de  $\mu$ -nilpotence que l'on a choisie, ces automates ne sont donc pas nécessairement  $\mu$ -nilpotents.

### 4.3 Construction

On va vouloir construire des automates cellulaires ayant des ensembles  $\mu$ -limite particuliers, par exemple de grande complexité ou avec des langages spécifiques. Pour ce faire, on propose dans cette partie une structure générique d'automate cellulaire que l'on pourra utiliser par la suite lors de diverses applications. Cet automate va en effet dépendre d'une machine de Turing qui sera propre à chaque application. On définit ici le fonctionnement général de ce type d'automates, ses propriétés, puis on donnera une caractérisation de l'ensemble  $\mu$ -limite obtenu en fonction de la machine de Turing utilisée.

L'idée générale est, en partant d'une configuration normale, de créer des zones de calcul de toutes tailles, que l'on appellera *segments*, séparées par un délimiteur spécial  $\#$ . On crée ainsi une succession de segments avec une distribution de tailles « aléatoire ». Un processus de *fusion* des segments entre eux permettra ensuite de faire disparaître les segments de petite taille en en créant d'autres plus grands. Ce processus ne sera jamais interrompu, de sorte qu'asymptotiquement, les délimiteurs disparaissent.

FIGURE 4.5 – L'automate  $\mathcal{A}$  sur une configuration normale : création des segments.

De cette manière, les mots qui conservent une grande densité dans les images successives d'une configuration normale (c'est à dire le langage  $\mu$ -limite d'après le lemme 4.1.3), sont les mots qui apparaissent à l'intérieur des segments. Ainsi, les mots que l'on écrit dans ces segments déterminent l'ensemble  $\mu$ -limite.

Dans chaque segment, une simulation de la machine aura lieu. On ne veut pas que cette machine ou les états qu'elle utilise pour calculer apparaissent dans l'ensemble  $\mu$ -limite. Pour cela, l'espace alloué spécifiquement au calcul de la machine sera négligeable par rapport à la taille du segment. Ainsi la densité de ces états tend vers 0 lorsque la taille des segments tend vers l'infini. La machine écrit dans tout le segment, un mot qui dépend du résultat de son calcul.

### 4.3.1 L'automate

#### Compteurs

On appelle  $\mathcal{A}$  l'automate que l'on construit, il est de rayon 1. On utilise la construction de compteurs de la partie 2.3, avec une modification : lorsque deux compteurs se rencontrent et qu'ils sont de même âge, ils disparaissent tous les deux mais la cellule sur laquelle, ils se sont rencontrés prend l'état  $\boxed{\#}$ . Le reste de la construction est inchangé : les compteurs effacent tout ce qu'ils rencontrent (y compris l'état  $\boxed{\#}$ ) sauf un compteur de même âge arrivant en sens inverse. On note  $v > 0$  la vitesse du bord intérieur des compteurs.

Dans une configuration normale, il existe toujours un état  $\boxed{*}$  à gauche et à droite de toute cellule, donc il y aura des départs de compteurs arbitrairement loin à gauche et à droite, on obtient donc le comportement illustré par la figure 4.5.

Le lemme suivant permet de localiser les  $\boxed{\#}$  créés par des compteurs issus de la configuration initiale :

**Lemme 4.3.1.** *Soit  $c \in X^{\mathbb{Z}}$  une configuration normale et  $(\gamma_k)_{k \in \mathbb{Z}}$  la suite croissante*

telle que :

- $\forall k \in \mathbb{Z}, c_{\gamma_k} = \boxed{*}$ ;
- $\forall x \in \mathbb{Z}$  tel que  $\forall k \in \mathbb{Z}, x \neq \gamma_k, c_x \neq \boxed{*}$ .

Alors on note  $(\beta_k = \lfloor \frac{1}{2}(\gamma_k + \gamma_{k-1}) \rfloor)_{k \in \mathbb{Z}}$  et :

- $\forall k \in \mathbb{Z}, \mathcal{A}^{\lfloor \frac{1}{2v}(\gamma_k - \gamma_{k-1}) \rfloor}(c)_{\beta_k} = \boxed{\#}$ ;
- $\forall k \in \mathbb{Z}, \forall x \in ]\beta_k, \beta_{k+1}[, \forall t \geq \frac{1}{v}|x - \gamma_k|, \mathcal{A}^t(c)_x \neq \boxed{\#}$ .

*Démonstration.* Avec une configuration normale, il existe des départs de compteurs arbitrairement loin de chaque côté, donc la suite  $(\gamma_k)_{k \in \mathbb{Z}}$  est bien indexée par  $\mathbb{Z}$ . La construction compteurs assure que lorsqu'une cellule est traversée par un compteur créé lors de l'étape initiale, elle ne sera plus traversée par un compteur. Donc en particulier, aucun  $\boxed{\#}$  ne pourra plus apparaître sur la cellule.

Il existe donc deux possibilités pour une cellule  $x$  :

- soit deux compteurs créés en  $\gamma_{k-1}$  et  $\gamma_k$  se rencontrent en  $\beta_k$ , auquel cas, après la phase de comparaison, ils donnent naissance à un état  $\boxed{\#}$ . Celui-ci apparaît donc à la fin de la comparaison lorsque les bords intérieurs des compteurs se rejoignent : au temps  $\lceil \frac{1}{2v}(\gamma_k - \gamma_{k-1}) \rceil$ .
- soit ce n'est pas le cas, et  $x$  est traversée par un compteur créé en  $\gamma_k$ . Dans ce cas,  $x$  ne contiendra plus de  $\boxed{\#}$  après le passage du bord intérieur du compteur en  $t = \frac{1}{v}|x - \gamma_k|$ .

□

## Segments

**Définition 4.3.1.** Soit  $c$  une configuration normale. Un intervalle  $[a, b] \subset \mathbb{Z}$  de cellules sera appelé *segment* entre les temps  $t_0$  et  $t_1 \in \mathbb{N}$  pour la configuration  $c$  lorsque :

- $\forall t \in [t_0, t_1], \mathcal{A}^t(c)_a = \mathcal{A}^t(c)_b = \boxed{\#}$ ;
- $\forall t \in [t_0, t_1], \forall x \in \mathbb{Z}$  si  $a < x < b$  alors  $\mathcal{A}^t(c)_x \neq \boxed{\#}$ .

La *longueur* ou *taille* du segment sera  $k = b - a$ .

*Remarque 4.3.1.* Un état  $\boxed{\#}$  peut exister pour deux raisons : il peut être présent sur la cellule où il se trouve depuis la configuration initiale et dans ce cas, il finira par être effacé par un compteur créé à l'étape initiale, ou il peut avoir été créé par la rencontre de compteurs de même âge. On ne considèrera dans la suite que les segments encadrés par des états  $\boxed{\#}$  provenant de rencontres de compteurs créés par un état  $\boxed{*}$  sur la configuration initiale.

Un segment étant un intervalle de cellules, on peut lui associer un mot pour chaque temps  $t \in \mathbb{N}$  :

**Définition 4.3.2.** Pour un segment  $s = [a, b]$  entre les temps  $t_0$  et  $t_1$ , on définit pour  $t \in [t_0, t_1]$  le mot contenu dans le segment au temps  $t$  :  $w_t(s) = \mathcal{A}^t(c)_{[a, b-1]}$ .

On aura besoin dans la suite que chaque segment connaisse le temps écoulé depuis la configuration initiale. Puisqu'un segment est encadré par deux  $\#$  issus de rencontres entre des compteurs connaissant ce temps, il suffit d'écrire à droite de chaque  $\#$  la valeur de ce temps codée en binaire. Ensuite, dans chaque segment, cette valeur sera incrémentée à chaque étape. Pour rendre cela possible, il faut que le segment soit assez grand pour contenir cette valeur. On voudra aussi éviter que les états utilisés pour mémoriser le temps apparaissent dans l'ensemble  $\mu$ -limite, et on souhaite donc que le temps soit écrit sur un espace de taille inférieure à  $\lfloor \sqrt{k} \rfloor$  où  $k$  est la taille du segment.

Pour avoir suffisamment de place, il faut donc que les segments soient de taille  $k$  avec  $\lfloor \log_2 t \rfloor \leq \lfloor \sqrt{k} \rfloor$ , c'est-à-dire  $k \geq \log_2(t)^2$ . Le processus de fusion que l'on décrira par la suite permettra d'assurer cette propriété, et les cas particuliers liés à deux états  $\boxtimes$  proches dans la configuration initiale sont traités avec un nombre fini d'états supplémentaires qui n'apparaissent plus au delà des premières étapes.

### Couleurs des segments

On note  $(\gamma_k)_k$  la suite des positions des  $\boxtimes$  sur la configuration initiale et  $(\beta_k)_k$  la suite des positions des  $\#$  formés par des rencontres de compteurs créés par ces  $\boxtimes$ . On dira qu'un segment est *initial* lorsqu'il est borné par les cellules  $\beta_k$  et  $\beta_{k+1}$  pour  $k \in \mathbb{Z}$ . Lors d'une fusion, un ou plusieurs  $\#$  successifs disparaissent, le segment qui en résulte est dit *successeur* et les segments qu'il remplace sont ses *prédécesseurs*.

On aura besoin dans ce qui suit d'attribuer des couleurs aux segments. On souhaite qu'à tout moment, la distribution des couleurs soit équiprobable. Pour cela on commence par attribuer des informations supplémentaires aux états  $\boxtimes$ . On remplace donc  $\boxtimes$  par quatre états contenant  $*_{R,0}$ ,  $*_{R,1}$ ,  $*_{B,0}$  et  $*_{B,1}$ .  $R$  et  $B$  sont les couleurs qui sont transmises des cellules  $\gamma_k$  aux segments commençant en  $\beta_k$ . Le bit d'information supplémentaire est envoyé lui au  $\#$  situé en  $\beta_k$ . Ainsi on remplace l'état  $\#$  par deux états contenant  $\#_0$  et  $\#_1$ . Le bit contenu par chaque  $\#$  sera utilisé lors des fusions. Lorsque l'on parle indifféremment de l'un ou l'autre de ces états, on notera de manière générique  $\boxtimes$  ou  $\#$ .

La figure 4.5 est remplacée par la figure 4.6. La couleur d'un segment est portée par ses extrémités uniquement et n'affecte pas les états au cœur du segment, mais pour les besoins de l'illustration, le segment entier est marqué.

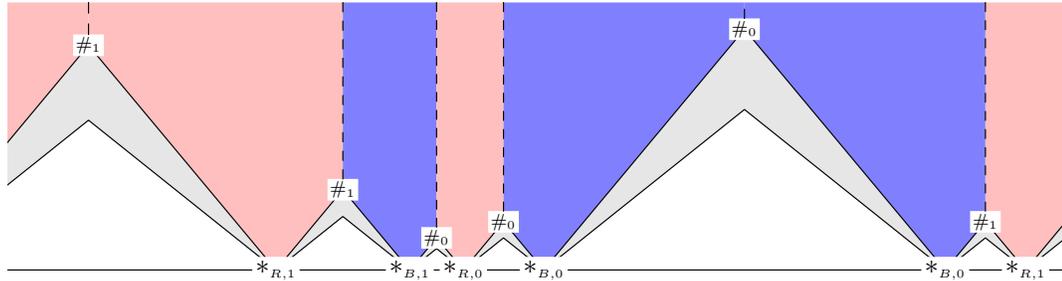


FIGURE 4.6 – L'automate  $\mathcal{A}$  sur une configuration normale : création des segments et attribution des couleurs.

### Fusion

Lorsque des segments deviennent trop petits (lorsque leur taille devient inférieure à  $(\lfloor \log_2 t \rfloor)^2$ ), ils doivent s'agrandir et donc fusionner avec au moins un voisin. Le calcul qui sera effectué dans les segments doit être indépendant du processus de fusion, on n'utilise donc pour la fusion que deux caractéristiques des segments : leur taille et leur couleur.

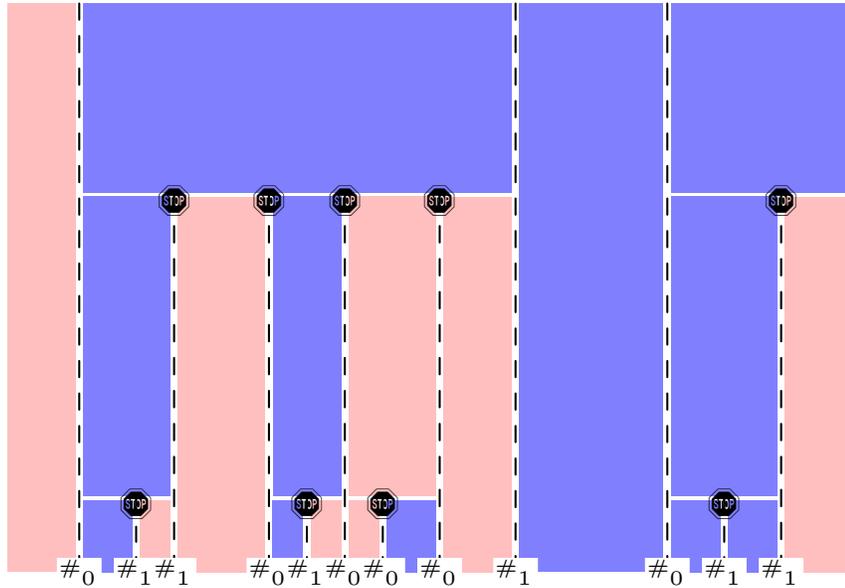
Les fusions ne se feront qu'aux temps  $(t_i = 2^i)_{i \in \mathbb{N}}$ , c'est à dire les temps auxquels l'écriture du temps en base 2 nécessite un bit de plus. Donc pour tout  $i \in \mathbb{N}$ , les segments existants auront  $2^{i+1} - 2^i = 2^i$  étapes pour se préparer.

Soit  $i \in \mathbb{N}$ , on s'intéresse à la fusion à l'étape  $t_i = 2^i$ , les segments peuvent donc se préparer entre les temps  $t_{i-1}$  et  $t_i$ . L'espace nécessaire pour écrire le temps devient alors  $i + 1$ , et les segments devront donc être plus grand que  $(i + 1)^2$ .

Soit un segment de taille  $k$ . On commence par envoyer des signaux de chaque côté du segment qui mesurent sa taille. Si le segment est trop petit, on pose une marque de chaque côté de ce segment pour avertir les voisins. Il suffit de  $(i + 1)^2$  étapes pour savoir si le segment est de taille supérieure à  $(i + 1)^2$ . Et en deux fois ce temps, on peut marquer les extrémités du segment.

Si un segment est suffisamment grand, il ne fait plus rien jusqu'à  $t_i$ . En revanche, les autres vont devoir fusionner avec un voisin, il reste donc à choisir lequel en suivant ces règles :

1. si aucun des deux voisins ne veut fusionner, la fusion se fait à gauche ;
2. si un seul des deux veut fusionner, la fusion se fait avec celui-ci ;
3. si les deux veulent fusionner, la fusion se fait avec un voisin de couleur différente si possible, sinon à gauche.

FIGURE 4.7 – L’automate  $\mathcal{A}$  sur une configuration normale : fusions de segments.

Lorsqu’un segment entre les cellules  $a$  et  $b$  veut fusionner avec un segment entre les cellules  $b$  et  $c$ , le  $\#$  en  $b$  disparaît à l’étape  $t_i$  ainsi que les compteurs qui mémorisaient l’âge depuis l’étape 0. De nouveaux segments sont donc créés entre les  $\#$  qui subsistent. Un segment successeur a donc deux ou plus prédécesseurs. Pour déterminer la couleur du nouveau segment, on utilise le bit du  $\#$  le plus à gauche ayant disparu à l’intérieur, si ce bit valait 0 le segment sera R, sinon B. La figure 4.7 illustre plusieurs étapes de fusion.

*Remarque 4.3.2.*

1. Grâce à ce procédé, au temps  $t \geq t_i$ , tous les segments sont plus grands que  $(i + 1)^2$ .
2. Si deux segments ont fusionné à  $t_i$ , au moins un des deux était plus petit que  $(i + 1)^2$ .
3. Si au moins trois segments ont fusionné à  $t_i$ , ils étaient tous plus petits que  $(i + 1)^2$ .

*Remarque 4.3.3.* Si l’on part d’une configuration normale, les couleurs des segments dépendent de la distribution initiale, donc la distribution des couleurs est équiprobable pour les segments initiaux. Ensuite, la couleur d’un segment successeur est choisie

en fonction d'un bit aléatoire qui n'est utilisé qu'une fois. Donc à tout moment la distribution des couleurs parmi les segments est équiprobable.

### Calcul

L'objet de cette partie n'est pas de décrire le calcul qui sera effectué dans les segments mais de contraindre ce calcul. La machine de Turing devra calculer en espace et en temps bornés de sorte que le calcul se fera dans la plupart des segments entre deux étapes de fusion. Ainsi, la fusion de deux segments n'interférera pas avec le calcul.

On montrera par la suite que presque tous les segments ont une taille inférieure à  $2^{i/4}$  entre les temps  $t_i$  et  $(t_{i+1} - 1)$  pour  $i \in \mathbb{N}$ . On veut que tous les calculs dans ces segments ou dans les segments qui respecteront cette condition à  $t_{i+1}$  soient terminés avant la prochaine étape de fusion en  $t_{i+1}$ . Le calcul débuté dans un segment de taille inférieure à  $2^{(i+1)/4}$  au temps  $t_i$ ,  $i \in \mathbb{N}$  doit être terminé au temps  $(t_{i+1} - 1)$ . Le calcul est considéré comme terminé lorsque la machine a effacé sa zone de calcul puis s'est elle-même effacée.

On note  $X$  l'alphabet de l'automate cellulaire  $\mathcal{A}$ , il est composé de :

- un alphabet d'écriture  $X_s$  qui est l'alphabet de sortie de la machine de Turing, on montrera que  $\Lambda_\mu(\mathcal{A}) \subseteq X_s^{\mathbb{Z}}$ ;
- un ensemble d'états  $X_1$  utilisés par la construction des compteurs et des segments, en particulier, les différents types d'états  $\boxtimes$  et  $\boxplus$ , tous les états des signaux des compteurs ou du calcul du temps;
- un alphabet de calcul  $X_2$  de la machine de Turing;
- certains produits cartésiens d'états des trois ensembles précédents.

La contrainte en espace permet d'éviter que les états qui apparaissent dans la zone de calcul apparaissent aussi dans le langage  $\mu$ -limite. Pour cela, la machine d'un segment de taille  $k$  disposera de  $\lfloor \log_2(k) \rfloor$  cellules adjacentes pour calculer. Ces cellules sont les plus à gauche du segment au début du calcul. Par la suite, elles peuvent être déplacées par la machine. Lorsqu'une cellule d'un segment fait partie de la zone de calcul, son état est le couple composé de l'état de  $X_s$  porté par la cellule et d'un état de  $X_2$ .

On contrôle aussi la phase d'écriture : dans chaque segment, au début du calcul, les cellules du segment contiennent l'état obtenu lors du calcul précédent. La machine fonctionne en « écriture seule » hors de sa zone de calcul, elle n'écrit un état de  $X_s$  qu'une seule fois dans chaque cellule. On requiert de plus que l'écriture se fasse de droite à gauche. Ainsi pour tout  $i \in \mathbb{N}$  et tout  $2^i \leq t < 2^{i+1}$ , un segment  $s$  a pour contenu  $w_t(s) = u_1 u_2$  où :

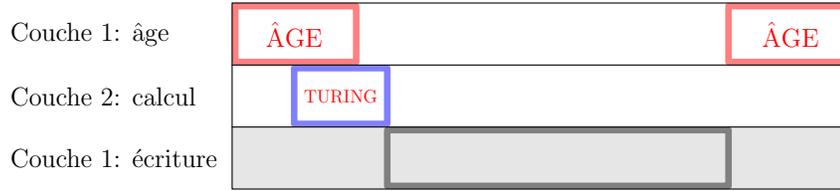


FIGURE 4.8 – Contenu d'un segment : superposition des trois couches qui le composent.

- $u_2$  est un suffixe de  $w_{2^i}(s)$  auquel peuvent être superposés des états de  $X_1$  et de  $X_2$  ;
- $u_1$  est un préfixe de  $w_{2^{i+1}-1}(s)$  auquel peuvent être superposés des états de  $X_1$  et de  $X_2$ .

### Segments acceptables

Comme dit précédemment, on s'intéresse en particulier à certains segments :

**Définition 4.3.3.** Pour  $i \in \mathbb{N}$ , un segment est dit *acceptable* entre les temps  $t_i$  et  $t_{i+1}$  lorsque sa taille  $k$  est telle que  $(i+1)^2 \leq k \leq 2^{i/4}$ .

On note  $S_t$  l'ensemble des segments acceptables au temps  $t$ .

La remarque 4.3.2 assure que la borne inférieure est respectée pour tous les segments, un segment est donc acceptable lorsqu'il n'est pas trop grand.

$S_t$  est, *a priori*, l'ensemble des segments acceptables dans l'orbite d'une configuration au moins. Cependant, lorsqu'on considère une configuration normale, tous les segments de  $S_t$  apparaîtront dans le diagramme espace-temps.

Le but est de montrer que les mots qui apparaissent dans le langage  $\mu$ -limite apparaissent dans les segments acceptables, c'est à dire que l'on peut se contenter de regarder les segments acceptables. On commence par prouver que la probabilité pour une cellule d'être dans un segment acceptable tend vers 1 au cours du temps. Cette probabilité porte sur l'ensemble des configurations, c'est à dire que l'on regarde la probabilité que, pour une configuration  $c$ , une cellule  $x$  et un temps  $t$ , le segment qui contient  $x$  au temps  $t$  soit acceptable. On a vu que l'ensemble des configurations normales est de mesure 1 dans l'ensemble des configurations.

**Lemme 4.3.2.** *Dans l'évolution d'une configuration choisie aléatoirement, pour toute cellule  $x$  :*

$$P [\exists s \in S_t, x \in s] \xrightarrow[t \rightarrow \infty]{} 1$$

*Démonstration.* Pour  $i \in \mathbb{N}$ , on se place au temps  $t_i \leq t < t_{i+1}$ . On choisit  $i \in \mathbb{N}$  assez grand tel que  $2^{i/4} - 2^{(i-1)/4} > i^2$  et  $2^{i/4} > 2i^2$ . Soit  $s$  un segment de taille  $k$  tel que  $k > 2^{i/4}$ . D'après la remarque 4.3.2,  $s$  est :

1. soit un segment initial ;
2. soit un segment successeur de deux segments exactement ;
3. soit un segment successeur d'au moins trois segments.

On commence par montrer que dans le cas 2, un prédécesseur était déjà non acceptable. On remonte ainsi jusqu'à trouver un prédécesseur non acceptable dans un des deux autres cas et on borne  $k$  en fonction de la taille de ce prédécesseur.

Dans le cas 2, un des deux prédécesseurs était plus petit que  $i^2$  d'après la remarque 4.3.2, donc l'autre était de taille supérieure à  $2^{i/4} - i^2 > 2^{(i-1)/4}$ , c'est à dire qu'il n'était pas acceptable à  $t_{i-1}$ . On peut ainsi constituer une chaîne de prédécesseurs qui étaient tous dans le cas 2 jusqu'à trouver un prédécesseur  $s'$  dans le cas 1 ou 3.

Un autre cas est possible : lorsque les deux prédécesseurs d'un segment non acceptable sont eux acceptables. C'est possible pour des temps suffisamment petits : à  $t_j$  avec  $2(j-1)^2 \geq 2^{j/4}$ . Les successeurs d'un tel segment deviennent vite acceptables puisque  $\sum_p p^2 = o(2^{p/4})$ . Donc, asymptotiquement, ce cas est négligeable.

On note alors  $h \in \mathbb{N}$  minimal tel que pour tout  $h < j \leq i$ ,  $s$  a un prédécesseur non acceptable à  $t_j$  dans le cas 2. On note  $l_j$  la taille de ces prédécesseurs. On a alors  $\forall j, h < j \leq i, l_{j+1} - l_j \leq j^2$ . On note  $l$  la taille de  $s'$ , prédécesseur non acceptable de  $s$  au temps  $t_h$ . Le segment  $s'$  était dans le cas 1 ou 3 et

$$k \leq l + \sum_{h \leq j \leq i} j^2 \leq l + \sum_{0 \leq j \leq i} j^2$$

Il existe  $i_0$  tel que pour tout  $i \geq i_0$ ,  $\sum_{0 \leq j \leq i} j^2 \leq 2^{(i-1)/4}$ . Ainsi, comme  $k \geq 2^{i/4}$ , on a  $l \geq 2^{i/4}(1 - 2^{-1/4})$ . Il s'ensuit que

$$k - l \leq l \left( \frac{1}{2^{1/4} - 1} \right)$$

C'est à dire que  $k \leq 7l$ .

Ainsi les segments qui ne sont pas acceptables proviennent des cas 1 et 3 et ont au plus septuplé de taille. On traite ces cas indépendamment.

**POINT 1:** Si  $\sigma$  est un segment de taille  $m$  au temps  $t_i \leq t < t_{i+1}$ , et qu'il est issu de la fusion d'au moins trois segments, alors sa probabilité d'apparaître à  $t_i$  était inférieure à  $\frac{4m}{i^2} 2^{-\frac{m}{i^2}}$ .

*Preuve:* Tous les prédécesseurs de  $\sigma$  étaient plus petits que  $i^2$  d'après la remarque 4.3.2. Donc  $\sigma$  a au moins  $p = \frac{m}{i^2}$  prédécesseurs. On regarde le mot formé par les

couleurs de ces segments, grâce aux règles de fusion, ce mot est d'une des formes suivantes :

$$R(RB)^a R^{p-2a-1}, (RB)^a R^{p-2a}, R(RB)^a B^{p-2a-1} \text{ ou } (RB)^a B^{p-2a}$$

(ou symétriquement en commençant par  $B$ ) pour un entier  $a \in \mathbb{N}$  tel que  $2a \leq p$ . Si ce n'était pas le cas, lors de la fusion au moins deux segments auraient été créés et pas un seul puisque les règles favorisent l'alternance ou à défaut la fusion à gauche. Une telle succession de segments est déterminée par le choix d'une des 8 formes et par  $a \leq p/2$ . En considérant que la répartition des couleurs est équiprobable comme dit en remarque 4.3.3, la probabilité d'apparition de cette suite est inférieure à  $4p2^{-p}$ . ◇ Point 1

POINT 2: *Un segment initial est de taille  $m$  avec une probabilité inférieure à*

$$m \left( \frac{4}{|X|} \right)^3 \left( \frac{|X| - 4}{|X|} \right)^{2m}$$

*Preuve:* Pour obtenir un segment initial de taille  $m$ , il faut trois états  $\boxed{*}$  distants de  $l_1$  puis  $l_2$ , avec  $l_1 + l_2 = 2m$ . En tenant compte des 4 états  $\boxed{*}$  différents, la probabilité d'avoir ces trois états sans autre  $\boxed{*}$  sur la configuration  $c$  est inférieure à

$$\frac{4}{|X|} \left( \frac{|X| - 4}{|X|} \right)^{l_1} \frac{4}{|X|} \left( \frac{|X| - 4}{|X|} \right)^{l_2} \frac{4}{|X|}$$

Avec les  $2m$  choix possibles pour  $l_1$ , on a une probabilité inférieure à

$$2m \left( \frac{4}{|X|} \right)^3 \left( \frac{|X| - 4}{|X|} \right)^{2m}$$

◇ Point 2

On note  $p_t$  la probabilité pour une cellule d'être hors d'un segment acceptable au temps  $t_i \leq t < t_{i+1}$ . La probabilité  $p_t = p_{t_i}$  est la somme des probabilités pour des cellules d'être dans des segments correspondant aux deux cas précédents et ayant au plus septuplé de taille. Pour un segment  $s$  de taille  $l$ , ce prédécesseur est noté  $s'$ , et comme  $s'$  a au plus septuplé de taille, il y a au plus  $6l/7$  emplacements différents pour  $s'$  par rapport à  $s$ . On note  $q = |X|$ .

$$p_{t_i} \leq \sum_{l \geq 2^{i/4-1}} l \times 6l/7 \left( 2l/7 \left( \frac{4}{q} \right)^3 \left( \frac{q-4}{q} \right)^{2l/7} \right) + \sum_{h \leq i} \sum_{l \geq 2^{i/4-1}} l \times 6l/7 \left( \frac{4l}{7h^2} 2^{-\frac{l}{7h^2}} \right)$$

$$p_{t_i} \leq \left( \frac{4}{q} \right)^3 \sum_{l \geq 2^{i/4-1}} \frac{12}{49} l^3 \left( \frac{q-4}{q} \right)^{2l/7} + i \sum_{l \geq 2^{i/4-1}} \frac{24}{49} l^3 \times 2^{-\frac{l}{7(i+1)^2}}$$

Et finalement,  $p_t \rightarrow_{t \rightarrow \infty} 0$ .

□

On peut maintenant prouver le lemme crucial : les mots qui persistent dans les segments acceptables aux temps  $t_i$  sont dans le langage  $\mu$ -limite, ceux qui disparaissent dans ces segments n'y sont pas. On utilisera plus tard un calcul qui donnera une distribution de probabilité sur un nombre fini de contenus possibles pour un segment de taille fixée. On inclut donc ce paramètre dans le lemme.

L'argument décisif est que le contenu d'un segment au temps  $t \in [t_i, t_{i+1} - 1]$  dépend de :

- sa taille ;
- des informations réparties de manière équiprobable entre tous les segments ;
- le contenu du segment au temps  $t_i$ .

Or au temps  $(t_{i+1} - 1)$ , dans tout segment acceptable, le calcul est fini, donc le segment a entièrement été réécrit et le contenu ne dépend plus du contenu au temps  $t_i$ . Ainsi, la distribution des contenus entre les segments acceptables de même taille au temps  $(t_{i+1} - 1)$  est équiprobable.

**Lemme 4.3.3.** *Soit  $c$  une configuration normale et  $u \in X^*$  :*

1. *si  $\exists \varepsilon > 0, \forall i_0 \in \mathbb{N}, \exists i \geq i_0, \forall l \in [(i+1)^2, 2^{i/4}]$ , on a*

$$P_{s \in S_{t_i}, |s|=l} \left[ d_{w_{t_{i+1}-1}(s)}(u) \geq \varepsilon \right] \geq \varepsilon$$

*alors  $u \in L_\mu(\mathcal{A})$  ;*

2. *si  $\forall \varepsilon > 0, \exists i_0 \in \mathbb{N}, \forall i \geq i_0, \forall l \in [(i+1)^2, 2^{i/4}]$ , on a*

$$P_{s \in S_{t_i}, |s|=l} \left[ d_{w_{t_{i+1}-1}(s)}(u) < \varepsilon \right] \geq 1 - \varepsilon$$

*alors  $u \notin L_\mu(\mathcal{A})$ .*

*Démonstration.* 1. Le premier point correspond aux mots qui apparaissent souvent dans des segments acceptables à des temps arbitrairement grands. On suppose ici que parmi les segments acceptables au temps  $(t_{i+1} - 1)$  de taille  $l$ , la probabilité de contenir  $u$  avec une densité supérieure à  $\varepsilon$  est infiniment souvent supérieure à  $\varepsilon$ . Dans ce cas, on montre que  $d_{\mathcal{A}^{t_i}(c)}(u) \not\rightarrow_{j \rightarrow \infty} 0$ . D'après le lemme précédent, puisqu'on s'intéresse à l'évolution de la configuration  $c$  normale, pour tout  $\delta > 0$ , il existe  $i_0 \in \mathbb{N}$  tel que  $\forall i \geq i_0, \forall x \in \mathbb{Z}, P \left[ x \in \cup_{s \in S_{t_i}} s \right] \geq 1 - \delta$ .

L'hypothèse nous permet de construire  $i_1 \geq i_0$  puis pour tout  $j \geq 1, i_{j+1} \geq i_j + 1$  tels que en notant  $\tau_j = t_{i_{j+1}} - 1$  :

$$\forall l \in [(i_j + 1)^2, 2^{i_j/4}], P_{s \in S_{t_{i_j}}, |s|=l} \left[ d_{w_{\tau_j}(s)}(u) \geq \varepsilon \right] \geq \varepsilon$$

Or, au temps  $\tau_j$ , le calcul est fini dans tout segment acceptable, donc le contenu du segment ne dépend plus du contenu à  $t_{i_j}$ , mais uniquement de la taille du segment et d'informations réparties de manière équiprobables. Donc, pour toute taille de segment, les contenus sont répartis de manière équiprobables entre les segments et :

$$\forall j \geq j_0, d_{\mathcal{A}^{\tau_j(c)}}(u) \geq (1 - \delta)\varepsilon^2$$

On prend  $\delta = 1/2$  et  $\forall j \geq j_0, d_{\mathcal{A}^{\tau_j(c)}}(u) \geq \varepsilon^2/2$

2. Le second point traite des mots qui tendent à disparaître des segments acceptables. On suppose ici que parmi les segments acceptables au temps  $(t_{i+1} - 1)$  de taille  $l$ , la probabilité de contenir  $u$  avec une densité inférieure à  $\varepsilon$  est toujours supérieure à  $1 - \varepsilon$ . Dans ce cas, on montre que  $d_{\mathcal{A}^t(c)}(u) \rightarrow_{t \rightarrow \infty} 0$ . Soit  $\varepsilon > 0$ . Soit  $i \in \mathbb{N}$ ,  $t \in [2^i, 2^{i+1}]$  et  $s \in S_t$ . On note  $k$  la taille de  $s$ . Tous les prédécesseurs de  $s$  sont de taille inférieure à  $2^{i/4}$  sinon  $s$  ne serait pas acceptable. Etant donné les contraintes imposées dans la partie 4.3.1, le calcul entre  $t_{i-1}$  et  $t_i - 1$  a eu le temps de finir dans chaque prédécesseur. Les contenus de ces segments à  $t_i - 1$  ont donc les mêmes propriétés que les segments acceptables. On sait aussi que l'on peut écrire  $w_t(s) = v_1 v_2$  où  $v_1$  et  $v_2$  sont des préfixe et suffixe de  $w_{2^{i+1}-1}(s)$  et  $w_{2^i}(s)$  modulo des superpositions avec des états de construction.

On choisit alors  $i_0$  tel que  $\forall l \in [(i+1)^2, 2^{i/4}]$  :

$$P_{s \in S_{t_i}, |s|=l} [d_{w_{t_{i+1}-1}(s)}(u) < \varepsilon] \geq 1 - \varepsilon$$

Les états de construction que l'on superpose aux mots de  $X_s^*$  sont :

- les états permettant de mémoriser le temps présents dans  $2\sqrt{k}$  cellules ;
- les états de calcul de la machine présents dans  $\log k$  cellules ;
- les délimiteurs des prédécesseurs de  $s$  sur lesquels la machine n'a pas encore réécrit, ainsi que les espaces utilisés par les prédécesseurs pour mémoriser le temps présents dans  $2\sqrt{l}$  cellules pour chacun.

Si on note  $(l_j)_j$  les longueurs des prédécesseurs, on majore la proportion de cellules concernées par

$$2\frac{\sqrt{k}}{k} + \frac{\log k}{k} + 2\frac{\sum_j \sqrt{l_j}}{\sum_j l_j}$$

Il existe  $i_1 \geq i_0 + 1$  tel que  $\forall k \geq (i_1 + 1)^2$  cette somme soit inférieure à  $\varepsilon$ .

On utilise de nouveau le fait que lorsque le calcul est fini dans tous les segments d'une taille donnée, la répartition des contenus parmi les segments est équiprobable. Elle l'est donc en particulier pour tous les prédécesseurs de  $s$ .

On prend  $i \geq i_1$ . Ainsi, comme  $i \geq i_0 + 1$ ,

$$P[d_{v_1}(u) < \varepsilon] \geq 1 - \varepsilon \text{ et } P[d_{v_2}(u) < \varepsilon] \geq 1 - \varepsilon$$

La densité moyenne de  $u$  dans un segment acceptable au temps  $t$  est inférieure à la somme des densités dans  $v_1$  et  $v_2$  auquel on ajoute  $\varepsilon$  pour les perturbations :

$$P[d_{v_1}(u) < \varepsilon]\varepsilon + (1 - P[d_{v_1}(u) < \varepsilon]) + \varepsilon = 2((1 - \varepsilon)\varepsilon + \varepsilon) + \varepsilon$$

Le lemme 4.3.2 dit que la proportion de segments acceptables est grande : il existe  $i_2 \geq i_1$  tel que  $\forall t \geq t_{i_2}, \forall x \in \mathbb{Z}, P[x \in \cup_{s \in S_t} s] \geq 1 - \varepsilon$ . Finalement, pour  $i \geq i_2$ , la densité de  $u$  dans  $\mathcal{A}^t(c)$  est donc inférieure à  $\varepsilon + 2((1 - \varepsilon)\varepsilon + \varepsilon) + \varepsilon$ .  $\square$

**Corollaire 4.3.4.**  $L_\mu(\mathcal{A}) \subseteq X_s^*$

On utilisera dans la suite ce lemme en ne décrivant que le résultat de la machine de Turing, c'est à dire en donnant le contenu des segments acceptables aux temps  $(t_i)_{i \in \mathbb{N}}$ . Dans la majorité des cas, le contenu du segment ne dépendra que de sa taille, la probabilité d'avoir ce contenu sera donc 1. Dans ce cas, on obtient la version simplifiée suivante :

**Lemme 4.3.5.** *Soit  $c$  une configuration normale et  $u \in X^*$  :*

1. *si  $\exists \varepsilon > 0, \forall i_0 \in \mathbb{N}, \exists i \geq i_0, \forall s \in S_{t_i}, d_{w_{t_{i+1}-1}(s)}(u) \geq \varepsilon$  alors  $u \in L_\mu \mathcal{A}$  ;*
2. *si  $\forall \varepsilon > 0, \exists i_0 \in \mathbb{N}, \forall i \geq i_0, \forall s \in S_{t_i}, d_{w_{t_{i+1}-1}(s)}(u) < \varepsilon$  alors  $u \notin L_\mu \mathcal{A}$ .*

## 4.4 Sous-shifts atteignables

Dans cette partie, on cherche à déterminer quels sous-shifts peuvent être l'ensemble  $\mu$ -limite d'un automate cellulaire. On commence par construire des automates cellulaires ayant des ensembles  $\mu$ -limite particuliers, puis on verra quelles contraintes doit respecter un ensemble pour être un ensemble  $\mu$ -limite potentiel.

### 4.4.1 Union de sous-shifts calculables

**Définition 4.4.1.** On dit qu'un sous-shift  $\Sigma \subseteq X_s^{\mathbb{Z}}$  est *calculable* par une machine de Turing  $M$  lorsque cette machine produit une demi-configuration *complète* pour ce sous-shift, c'est à dire :

- l'alphabet de  $M$  contient  $X_s$  ;
- sur une entrée vide,  $M$  produit une demi-configuration  $c \in X_s^{\mathbb{N}}$  telle que  $\mathcal{L}(\Sigma) = \mathcal{L}(c)$  et pour tout mot  $u \in \mathcal{L}(c)$ ,  $\liminf_{n \rightarrow +\infty} d_{c_{[0,n]}}(u) > 0$  ;
- lorsqu'un symbole de  $X_s$  est écrit sur le ruban, il n'est jamais plus effacé.

La dernière condition permet d'obtenir un mot du langage de  $\Sigma$  en arrêtant la machine à n'importe quel moment. En effet tous les préfixes du mot inscrit sur le ruban qui sont des mots de  $X_s^*$  sont des mots du langage de  $\Sigma$ .

On montre maintenant que ces sous-shifts peuvent tous être des ensembles  $\mu$ -limite d'automates cellulaires, ainsi que toutes les unions récursivement énumérables de tels sous-shifts.

**Théorème 4.4.1.** *Soit  $Q$  un alphabet fini et  $(\Sigma_i)_{i \in \mathbb{N}}$  une famille récursivement énumérable de sous-shifts calculables (c'est à dire que la famille des machines calculant les sous-shifts est récursivement énumérable) sur l'alphabet  $Q$ , il existe un automate cellulaire dont l'ensemble  $\mu$ -limite est exactement  $\overline{\bigcup_{i \in \mathbb{N}} \Sigma_i}$ .*

*Démonstration.* Il existe une machine de Turing  $M$  qui énumère des machines  $(M_i)_i$  telles que  $M_i$  calcule le sous-shift  $\Sigma_i$ . On note  $c_i$  la demi-configuration complète associée à  $\Sigma_i$  et  $Q$  l'union des alphabets des sous-shifts.

On va utiliser l'automate  $\mathcal{A}$  construit précédemment. On décrit le calcul effectué dans un segment  $s$  de taille  $k$  entre les temps  $t_i$  et  $t_{i+1} - 1$  pour  $i \in \mathbb{N}$ . Le temps de calcul total autorisé est  $2^i$ .

- On commence par calculer la taille du segment en  $2k$  étapes
- Dans les  $(\log k)$  premières cellules, on simule  $M$  pour obtenir les descriptions des  $j$  premières machines. On prend  $j$  maximal tel que le calcul soit borné en espace  $(\log k)$  et en temps de manière à ce que cette étape soit terminée au temps  $(t_i + t_{i+1})/2 = 2^i + 2^{i-1}$ .
- On découpe ensuite les  $(\log k)$  premières cellules en  $j$  zones de calcul de même taille et on simule chacune des  $M_a$ ,  $a < j$  dans une de ces aires. Ce calcul termine par manque d'espace ou au plus tard en  $t_{i+1} - 1 - 2k$ .
- La simulation de  $M_a$  produit un préfixe  $w_a$  de  $c_a$ .
- On écrit ensuite dans  $s$  de la manière suivante :
  - On découpe  $\sqrt{k}$  fragments de taille  $\sqrt{k}$ . Ces fragments sont séparés par un symbole  $\$1$ .
  - On attribue un fragment sur deux au sous-shift  $\Sigma_0$  puis un sur quatre au sous-shift  $\Sigma_1$  et ainsi de suite jusqu'à attribuer un fragment sur  $2^{a+1}$  à  $\Sigma_a$ .
  - Dans un fragment attribué à  $\Sigma_a$ , on écrit des copies de  $w_a$  séparées par un symbole  $\$2$ .
  - La durée totale de l'écriture à laquelle on ajoute le temps nécessaire à la machine pour effacer sa zone de calcul est inférieure à  $2k$ .

Le principe est que dans des segments suffisamment grands, les mots  $w_a$  produits seront grands aussi, ainsi, les symboles  $\$1$  et  $\$2$  auront une faible densité alors que tout mot du sous-shift  $\Sigma_a$  finira par apparaître avec une densité non nulle dans  $w_a$  et donc dans les segments.

POINT 1: Soit  $u \in \left( X^* \setminus \left( \bigcup_{j \in \mathbb{N}} \mathcal{L}(\Sigma_j) \right) \right)$ ,  $u \notin L_\mu(\mathcal{A})$ .

*Preuve:* Le corollaire 4.3.4 assure que  $L_\mu(\mathcal{A}) \subseteq X_s^*$ . Et  $X_s = Q \cup \{\$1, \$2\}$ . On va donc prouver que  $\$1$  et  $\$2$  ne sont pas dans  $L_\mu(\mathcal{A})$ .

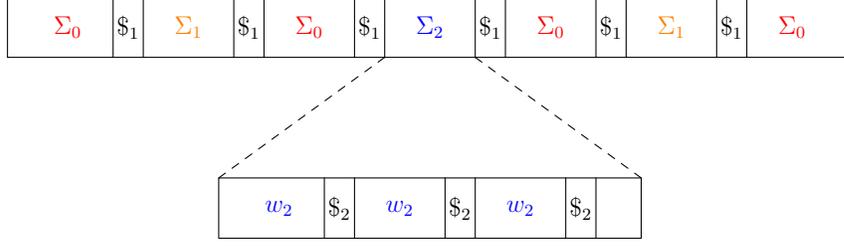


FIGURE 4.9 – Contenu de la couche d'écriture d'un segment.

Soit  $c$  une configuration normale. Pour tout  $\varepsilon > 0$ , il existe  $i_0$  tel que  $\forall i \geq i_0$ ,  $\sqrt{(i+1)^2} \geq \frac{1}{\varepsilon}$ . Donc  $\frac{1}{\sqrt{(i+1)^2}} \leq \varepsilon$ . Or pour tout segment  $s \in S_{t_i}$ , lorsque le calcul est terminé dans  $s$ , il y a au plus un  $\$1$  toutes les  $\sqrt{(i+1)^2}$  cellules. Donc

$$d_{\mathcal{A}^{t_{i+1}-1}(c)}(\$1) \leq \frac{1}{\sqrt{(i+1)^2}} \leq \varepsilon$$

Et donc, le cas 2 du lemme 4.3.5 s'applique et  $\$1 \notin L_\mu(\mathcal{A})$ .

Pour  $\$2$ , on fait exactement le même raisonnement, en prenant  $i_1 \in \mathbb{N}$  tel que pour tout  $i \geq i_1$ , et dans tout segment  $s \in S_{t_i}$ ,  $\forall j \leq 1/\varepsilon$ ,  $|w_j| \geq 1/\varepsilon$ . Ainsi, pour les fragments alloués aux  $\lfloor 1/\varepsilon \rfloor$  premiers sous-shifts, la densité de  $\$2$  est inférieure à  $\varepsilon$  et les autres fragments occupent une proportion  $\frac{1}{2^{1/\varepsilon}}$  de  $s$ . Donc

$$d_{\mathcal{A}^{t_{i+1}-1}(c)}(\$2) \leq \varepsilon + \frac{1}{2^{1/\varepsilon}}$$

De nouveau, le cas 2 du lemme 4.3.5 permet de conclure.

Si maintenant  $u \in X^* \setminus \left( \bigcup_{j \in \mathbb{N}} \mathcal{L}(\Sigma_j) \right)$ ,  $u$  n'apparaît dans aucun  $w_j$  donc

$$\forall i \in \mathbb{N}, \forall s \in S_{t_i}, d_{\mathcal{A}^{t_{i+1}-1}(c)}(u) = 0$$

On conclut encore avec le même résultat. ◇ Point 1

POINT 2: Soit  $u \in \bigcup_{j \in \mathbb{N}} \mathcal{L}(\Sigma_j)$ ,  $u \in L_\mu(\mathcal{A})$ .

*Preuve:* Cette fois, on va utiliser le cas 1 du lemme 4.3.5. Il existe  $j \in \mathbb{N}$  tel que  $u \in \mathcal{L}(\Sigma_j)$ . Il existe  $i_0 \in \mathbb{N}$  tel que pour tout  $i \geq i_0$ , dans tout segment  $s \in S_{t_i}$  la machine  $M_j$  est produite par  $M$ , et calcule un préfixe de  $c_j \in \mathcal{L}(\Sigma_j)^\mathbb{N}$ . Comme  $c_j$  est complète pour  $\Sigma_j$ , il existe  $\varepsilon > 0$  et  $n_0 \in \mathbb{N}$  tel que  $\forall n \geq n_0$ ,  $d_{c_j[0,n]}(u) > \varepsilon$ . Il existe aussi  $i_1 \geq i_0$  tel que  $\forall i \geq i_1$ ,  $\forall s \in S_{t_i}$ , le préfixe de  $c_j$  produit dans  $s$  soit de longueur supérieure à  $n$ .

Dans un segment  $s$  acceptable à  $(t_{i+1} - 1)$ , il y a donc une proportion supérieure à  $\frac{1}{2^{j+2}}$  du segment dédiée au sous-shift  $\Sigma_j$ . Une proportion  $\frac{1}{i+1}$  du segment est occupée par les états  $\mathbb{S}_1$  et  $\frac{1}{n}$  par les états  $\mathbb{S}_2$ . Finalement :

$$d_{w_{t_{i+1}-1}(s)}(u) \geq \frac{\varepsilon}{2^{j+1}} \left(1 - \frac{1}{i+1}\right) \left(1 - \frac{1}{n}\right)$$

Ainsi  $d_{w_{t_{i+1}-1}(s)}(u) \not\rightarrow_{i \rightarrow \infty} 0$ .

◇ Point 2

□

#### 4.4.2 Applications

On donne ici deux classes de sous-shifts pour lesquels on peut construire une configuration générique.

##### Applications aux substitutions primitives

**Définition 4.4.2.** On appelle *substitution* une fonction  $\tau : X^* \rightarrow X^* \setminus \{\varepsilon\}$  définie sur l'alphabet  $X$  et étendue à  $X^*$  avec la propriété :

$$\forall u, u' \in X^*, \tau(uu') = \tau(u)\tau(u')$$

On appelle *point fixe* d'une substitution une demi-configuration  $c \in X^{\mathbb{N}}$  telle que  $\tau(c) = c$ .

La substitution la plus célèbre est celle dite de *Fibonacci* sur l'alphabet  $\{0, 1\}$  définie comme suit :

- $F(0) = 1$  ;
- $F(1) = 10$ .

Son unique point fixe est la demi-configuration  $\tau^\infty(1) = 1011010110110\dots$

**Définition 4.4.3.** On dit de plus qu'une substitution est *primitive* lorsqu'il existe  $k \in \mathbb{N}$  tel que pour toutes lettres  $a, b \in X$ ,  $a$  apparaît dans  $\tau^k(b)$ .

Pour toute substitution primitive  $\tau$ , il existe  $i \in \mathbb{N}$  et une lettre  $a \in X$  tels que  $\tau^i(a) \in aX^*$ . On peut alors construire un point fixe  $c_{\tau,a}$  dont tous les  $\tau^{ni}(a)$  sont préfixes pour  $n \in \mathbb{N}$ .

On peut montrer [13] que tout sous mot  $u$  de  $c_{\tau,a}$  apparaît avec une densité comprise entre  $\frac{\alpha}{|X|^{|u|}}$  et  $\frac{\beta}{|X|^{|u|}}$  dans  $c_{\tau,a}$  (pour deux constantes  $\alpha$  et  $\beta$ ). Ainsi  $c_{\tau,a}$  est générique pour le sous-shift  $\Sigma_{\tau,a}$  qu'elle engendre. Le théorème 4.4.1 nous dit alors qu'il existe un automate cellulaire dont l'ensemble  $\mu$ -limite est  $\Sigma_{\tau,a}$ .

### Sous-shifts sofiques transitifs

**Définition 4.4.4.** Un sous-shift  $\Sigma$  sur l'alphabet  $X$  est dit *transitif* lorsque pour tous mots  $u, v \in \mathcal{L}(\Sigma)$ , il existe  $w \in X^*$  tel que  $uwv \in \mathcal{L}(\Sigma)$ .

On montre que tous les sous-shifts sofiques transitifs sont des ensembles  $\mu$ -limite.

**Proposition 4.4.2.** Soit  $\Sigma$  un sous-shift sofique transitif, alors il existe un automate cellulaire  $\mathcal{A}$  tel que  $\Lambda_\mu(\mathcal{A}) = \Sigma$ .

*Démonstration.* On rappelle qu'un sous-shift est sofique lorsque son langage est le langage de sortie d'un transducteur. Il a été prouvé [24, 3.3.11] que les sous-shifts sofiques transitifs sont exactement ceux pour lesquels le graphe associé au transducteur peut être choisi fortement connexe. On va construire une demi-configuration complète pour  $\Sigma$ .

Il existe donc un graphe  $G$  fortement connexe associé à  $\Sigma$ , les arrêtes de  $G$  étant indexées par des lettre de  $X$ . A chaque chemin dans  $G$ , on fait donc correspondre un mot de  $\mathcal{L}(\Sigma)$ . On particularise un état  $s$  de  $G$  et on s'intéresse à l'ensemble des cycles de  $s$  à  $s$  dans le graphe. Pour plus de simplicité, on suppose qu'il existe des cycles de toute taille. Si ce n'est pas le cas, il faut utiliser le pgcd des longueurs des cycles : à partir d'une taille minimale, il existe des cycles de taille toutes multiple du pgcd.

Pour  $k \in \mathbb{N}$ , on note  $u_k$  un mot correspondant à un cycle de taille minimale commençant et finissant en  $s$  et contenant tous les mots de taille  $k$  de  $\mathcal{L}(\Sigma)$ . On définit alors la demi-configuration  $c = u_1 w_0 u_1^2 w_1 u_1^4 w_2 \dots$  en concaténant les facteurs  $u_1^{2^i} w_i$  avec  $|w_i| = 2^i |u_1|$ . On donnera plus tard le contenu des mots  $w_i$ . Ainsi, pour tout mot  $u \in \mathcal{L}(\Sigma)$  de taille 1, on a  $\forall n \in \mathbb{N}, d_{c_{[0,n]}}(u) \geq d_{c_{[0,n]}}(u_1) d_{u_1}(u) \geq \frac{1}{3} d_{u_1}(u) > 0$ . En effet, comme la moitié de la configuration contient le mot  $u_1$ , la densité de  $u_1$  dans un préfixe de la demi-configuration est toujours proche de  $1/2$ . Ainsi  $\liminf_{n \rightarrow \infty} d_{c_{[0,n]}}(u) > 0$ . On va répéter ce procédé avec les mots de taille 2 dans les  $w_j$ .

Il existe  $n \in \mathbb{N}$ , tel que  $|w_n| \geq |u_2|$ . Donc il existe un cycle  $v_2$  de taille  $|w_n|$  contenant tous les mots de taille 2, en prenant  $v_2 = u_2 v'_2$  avec  $v'_2 \in X^*$ . On pose alors pour  $j \in \mathbb{N}$ ,  $w_{n+2j} = v_2^{2^{2j}}$ . On a donc bien

$$|w_{n+2j}| = |v_2^{2^{2j}}| = 2^{2j} |v_2| = 2^{n+2j} |u_1|$$

Si par exemple  $n = 2$ , on a  $c = u_1 w_1 u_1^2 v_2 u_1^4 w_3 u_1^8 v_2^4 \dots$ . Ainsi pour tout mot  $u$  de taille 2, on a  $d_{c_{[0,n]}}(u) \geq d_{c_{[0,n]}}(v_2) d_{v_2}(u) \geq \alpha d_{v_2}(u) > 0$  pour une constante  $\alpha > 0$ . Ainsi  $\liminf_{n \rightarrow \infty} d_{c_{[0,n]}}(u) > 0$ . Pour  $i < n$ , on pose alors  $w_i = v_1^{2^i}$ .

On continue la construction avec  $u_k$  pour les mots de taille  $k$  en utilisant à chaque fois la moitié des  $w_i$  restant. De cette manière, pour tout mot  $u$  de  $\mathcal{L}(\Sigma)$ , on a  $\liminf_{n \rightarrow \infty} d_{c_{[0,n]}}(u) > 0$ , et  $c$  est bien une demi-configuration complète pour  $\Sigma$ .

Le théorème 4.4.1 permet de conclure : il existe un automate cellulaire dont l'ensemble  $\mu$ -limite est  $\Sigma$ .  $\square$

### 4.4.3 Vers une caractérisation ?

La question naturelle que l'on peut se poser est celle de la caractérisation exacte des sous-shifts qui peuvent être ensembles  $\mu$ -limite d'un automate cellulaire. Le théorème 4.4.1 montre qu'une large famille de sous-shifts peut être obtenue. On encadre dans cette partie l'ensemble des sous-shifts atteignables.

#### Du point de vue de la calculabilité

On commence par borner la complexité d'un ensemble  $\mu$ -limite :

**Lemme 4.4.3.** *Soit  $\mathcal{A}$  un automate cellulaire,  $\Lambda_\mu(\mathcal{A})$  est au plus  $\Sigma_3$  dans la hiérarchie arithmétique.*

En effet, on peut exprimer cette propriété de la manière suivante pour un mot  $u \in X^*$  :

$$u \in \Lambda_\mu(\mathcal{A}) \Leftrightarrow \exists \varepsilon > 0, \forall t \in \mathbb{N}, \exists t' \geq t, \mu(\mathcal{A}^{-t'}([u])) \geq \varepsilon$$

On ne sait pas encore si cette borne est atteinte, c'est à dire s'il existe un automate cellulaire dont l'ensemble  $\mu$ -limite est de complexité  $\Sigma_3$ , en revanche, on va montrer que l'on peut s'en rapprocher.

**Lemme 4.4.4.** *Il existe un automate  $\mathcal{A}$  tel que  $\Lambda_\mu(\mathcal{A})$  est  $\Sigma_2$ -dur.*

*Démonstration.* On va construire un automate cellulaire pour lequel déterminer l'appartenance à son ensemble  $\mu$ -limite est au moins aussi difficile que de déterminer si une machine de Turing est totale (c'est à dire si elle s'arrête sur toute entrée). Ce problème est  $\Pi_2$ -dur [30].

On va utiliser le lemme 4.3.3, donc on décrit le calcul de la machine de Turing  $M$  dans un segment  $s$  acceptable au temps  $t_{n+1} - 1$ . On prendra l'alphabet d'écriture  $X_s = \{0, 1, \$_0, \$_1, \$_2\}$ . On utilise une bijection calculable  $\lambda$  de  $\mathbb{N}$  dans  $\mathbb{N}^2$  pour créer une fonction  $f : \mathbb{N} \rightarrow X_s^*$  telle que  $\forall k \in \mathbb{N}, f(k) = \$_0 b_{i,j} \$_1 i \$_2 j$  où  $(i, j) = \lambda(k)$  et  $b_{i,j} \in \{0, 1\}$ . Pour commencer, tous les  $b_{i,j}$  valent 0. On utilise une énumération des machines de Turing. L'idée est d'énumérer les  $f(k)$  et de mettre  $b_{i,j}$  à 1 si la machine  $i$  s'arrête sur l'entrée  $j$ .

La machine  $M$  commence par écrire un préfixe de

$$c = f(0)f(1)f(0)f(2)f(0)f(1)f(0)f(3)\dots$$

dans le segment  $s$ . A priori, un tel préfixe a déjà été écrit pour un des prédécesseurs de  $s$ , il suffit donc de le compléter. Si des  $b_{i,j}$  sont déjà à 1, ils le restent. Cela prend un temps linéaire en la taille du segment.

Pendant le temps restant  $\tau$  (au moins polynomial en la taille du segment),  $M$  teste les  $n$  premiers couples  $i, j$  tels que  $b_{i,j} = 0$ . C'est à dire que pour chacun de ces couples,  $M$  simule la machine  $i$  sur l'entrée  $j$  pendant  $\tau/n$  étapes sur son espace de travail (logarithmique en la taille du segment). Lorsqu'une simulation se termine, toutes les occurrences du bit  $b_{i,j}$  correspondant sont mises à 1.

Pour tout couple  $i, j \in \mathbb{N}^2$ , la densité du mot  $u_k = b_{i,j}i\$2j\$0$  (pour  $(i, j) = \lambda(k)$ ) tend vers une densité non nulle lorsque  $n$  croît. En effet on peut prendre  $\lambda$  tel que  $|u_k| \leq 2 \log k$ . On pose alors  $v_n$  le préfixe de  $c$  qui s'arrête à la première apparition de  $f(n)$ .

$$d_{v_n}(u_k) \geq \frac{2^{n-k}}{\sum_{l \leq n} 2^{n-l} 2 \log l} \geq \frac{\alpha}{2^k}$$

pour une constante  $\alpha > 0$ .

Comme de plus

$$\frac{\sum_{l \leq n} 2^{n-l} 2 \log l}{\sum_{l \leq n+1} 2^{n+1-l} 2 \log l} \xrightarrow{n \rightarrow +\infty} \frac{1}{2}$$

on a :

$$\forall N \in \mathbb{N}, d_{c_{[0, N]}}(u_k) \geq \alpha \beta 2^{-k} > 0$$

pour une constante  $\beta > 0$ .

Ainsi, si la machine n'est pas totale, la limite inférieure de la densité du mot  $0\$1i\$2$  sera strictement positive. Si par contre la machine est totale, la densité de  $0\$1i\$2$  va tendre vers 0. En effet, dans tous les segments assez grands, la simulation pour les couples  $(i, j), j \leq j_0$ , pour un certain  $j_0$ , aura eu le temps de finir, et la densité de  $0\$1i\$2$  sera inférieure à  $\alpha \beta 2^{-k_0}$  où  $f(k_0) = \$0b_{i, j_0}\$1i\$2j_0$ .

Finalement, pour tout  $i \in \mathbb{N}$ ,  $0\$1i \in L_\mu(\mathcal{A}) \Leftrightarrow$  la machine  $i$  n'est pas totale.  $\square$

### Sous-shifts inatteignables

La caractérisation du langage  $\mu$ -limite d'un automate cellulaire par les densités dans les images des configurations normales donne une contrainte sur les mots de ce langage. En effet, pour tout mot de  $L_\mu(\mathcal{A})$ , il existe nécessairement une configuration contenant un nombre arbitraire d'occurrences de ce mot.

**Définition 4.4.5.** Un sous-shift  $\Sigma$  sur l'alphabet  $X$  est dit *non-errant* si et seulement si

$$\forall u \in \mathcal{L}(\Sigma), \exists v \in X^*, uvu \in \mathcal{L}(\Sigma)$$

**Proposition 4.4.5.** Soit  $\mathcal{A}$  un automate cellulaire, alors  $\Lambda_\mu(\mathcal{A})$  est non-errant.

*Démonstration.* On suppose le contraire, il existe donc  $v \in L_\mu(\mathcal{A})$  tel que

$$\forall w \in X^*, vvw \notin L_\mu(\mathcal{A})$$

Soit  $c$  une configuration normale, d'après le lemme 4.1.3, il existe  $(t_n)_n$  et  $\varepsilon > 0$  tels que  $d_{\mathcal{A}^{t_n}(c)}(v) > \varepsilon$ .

On choisit alors  $k \in \mathbb{N}$  tel que  $\frac{|v|}{|v|+k} \leq \frac{\varepsilon}{2}$ . On a supposé que

$$\left( \bigcup_{j \leq k} vX^jv \right) \cap L_\mu(\mathcal{A}) = \emptyset$$

Donc

$$\forall w \in \bigcup_{j \leq k} vX^jv, d_{\mathcal{A}^{t_n}(c)}(w) \rightarrow_{t \rightarrow \infty} 0$$

Donc il existe  $t_0$  tel que  $\forall t \geq t_0, |v|d_{\mathcal{A}^t(c)}(\bigcup_{j \leq k} vX^jv) \leq \frac{\varepsilon}{2}$ .

On note  $X_v^q$  l'ensemble des mots de  $X^q$  qui ne contiennent pas  $v$  comme sous mot, en particulier, pour  $q < |v|$ ,  $X_v^q = X^q$ . Pour tout  $t \geq t_0$ ,

$$d_{\mathcal{A}^t(c)}(v) = |v| \sum_{i \in \mathbb{N}} \sum_{w \in X_v^i} d_{\mathcal{A}^t(c)}(vw) \leq |v| \sum_{i \leq k} \sum_{w \in X_v^i} d_{\mathcal{A}^t(c)}(vw) + |v| \sum_{i > k} \sum_{w \in X_v^i} d_{\mathcal{A}^t(c)}(vw)$$

Donc  $d_{\mathcal{A}^t(c)}(v) \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2}$ , ce qui contredit l'hypothèse.  $\square$

En particulier, cette proposition permet de prouver que l'ensemble limite de l'automate MAX ne peut être l'ensemble  $\mu$ -limite d'un automate cellulaire. On rappelle

$$\Lambda(\text{MAX}) = (\mathbb{N}^{\square\square^*\square^{\mathbb{N}}}) \cup (\square^{\mathbb{Z}}) \cup (\mathbb{N}^{\square\square^{\mathbb{N}}}) \cup (\mathbb{N}^{\square\square^{\mathbb{N}}})$$

Le mot  $\square\square\square$  apparaît au plus une fois dans une configuration de  $\Lambda(\text{MAX})$ .

En utilisant ce lemme de manière répétée, on peut montrer le résultat suivant :

**Corollaire 4.4.6.** Pour tout mot  $u \in L_\mu(\mathcal{A})$ , il existe  $c \in \Lambda_\mu(\mathcal{A})$  tel que  $c$  contienne une infinité d'occurrences de  $u$ .

*Démonstration.* On construit une suite  $(v_i)_{i \in \mathbb{N}}$  de mots de  $X^*$  tels que  $\forall i \in \mathbb{N}, |v_i|_u \geq 2^i$  et  $v_i$  est un préfixe de  $v_{i+1}$ . On pose  $v_0 = u$ . D'après la proposition 4.4.5 affirme qu'il existe  $w_0 \in X^*$  tel que  $v_0 w_0 v_0 \in L_\mu(\mathcal{A})$ . On pose alors  $v_1 = v_0 w_0 v_0$ .

Par récurrence, en utilisant la proposition 4.4.5, on montre que pour tout  $i \in \mathbb{N}$ , il existe  $w_i \in X^*$  tel que  $v_i w_i v_i \in L_\mu(\mathcal{A})$  et on pose  $v_{i+1} = v_i w_i v_i$ .

Pour tout  $i \in \mathbb{N}$ , il existe donc une configuration  $c_i \in [v_i]_0$ . On peut extraire de la suite  $(c_i)_i$  une suite convergente vers  $c \in \Lambda_\mu(\mathcal{A})$ . Pour tout  $i \in \mathbb{N}$ , il existe  $j \in \mathbb{N}$  tel que  $c_{[0, |v_i| - 1]} = c_{j_{[0, |v_i| - 1]}}$ . Donc  $c$  contient une infinité d'occurrences de  $u$ .  $\square$

Dans le cas des sous-shifts sofiques, on utilise la proposition suivante pour décrire les ensembles  $\mu$ -limite sofiques :

**Proposition 4.4.7.** *Le langage de tout sous-shift sofique non-errant est une union finie de langages de sous-shifts transitifs.*

*Démonstration.* Soit  $\Sigma$  un sous-shift sofique non-errant et  $G$  le graphe d'un transducteur  $f$  sur l'alphabet  $X'$  calculant  $\Sigma$ . On note  $n$  le nombre de sommets de  $G$ . On décompose  $G$  en un nombre fini de composantes fortement connexes  $(C_i)_{i \leq K}$ . On note  $\Sigma_i$  le sous-shift associé à  $C_i$ .

Soit  $u \in \mathcal{L}(\Sigma)$ . Comme  $\Sigma$  est non-errant, il existe une configuration  $c \in \Sigma$  contenant  $(n + 1)$  occurrences de  $u$ . Il existe donc  $c' \in X'^*$  telle que  $f(c') = c$ . Lorsque  $c$  est calculée par le transducteur, chaque occurrence de  $u$  est produite à partir d'un état de  $G$ . On a donc distingué  $(n + 1)$  états de  $G$  qui n'a que  $n$  états, ce qui signifie qu'il existe un cycle dans le graphe qui produit un mot contenant  $u$ . Ce cycle ne peut appartenir qu'à une composante fortement connexe, et donc il existe  $i \leq K$  tel que  $u \in \mathcal{L}(\Sigma_i)$ . Finalement

$$\mathcal{L}(\Sigma) = \bigcup_{i \leq K} \mathcal{L}(\Sigma_i)$$

$\square$

En utilisant la proposition 4.4.2, on peut maintenant caractériser les sous-shifts sofiques réalisables en tant qu'ensembles  $\mu$ -limite.

**Théorème 4.4.8.** *Les sous-shifts sofiques qui sont ensemble  $\mu$ -limite d'un automate cellulaire sont exactement les sous-shifts sofiques non-errants.*

*Démonstration.* Soit  $\mathcal{A}$  un automate cellulaire dont l'ensemble  $\mu$ -limite  $\Lambda_\mu(\mathcal{A})$  est sofique. D'après la proposition 4.4.5,  $\Lambda_\mu(\mathcal{A})$  est non-errant.

Soit  $\Sigma$  un sous-shift sofique non-errant. La proposition 4.4.7 affirme qu'on peut l'écrire comme l'union finie de sous-shifts transitifs :

$$\Sigma = \bigcup_{i \leq K} \Sigma_i$$

On a montré dans la preuve de la proposition 4.4.2 qu'un sous-shift transitif est calculable. On conclut donc en utilisant le théorème 4.4.1 avec la famille de sous-shifts transitifs  $(\Sigma_i)_{i \leq K}$ .  $\square$

## 4.5 Théorème de Rice

Dans cette partie, on s'intéresse aux propriétés des ensembles  $\mu$ -limite d'automates cellulaires. Par exemple, la  $\mu$ -nilpotence est une propriété d'ensembles  $\mu$ -limite qui a été démontrée indécidable [4]. Dans le cas des ensembles limite, J. Kari a prouvé [20] un équivalent du théorème de Rice, c'est à dire que toute propriété non triviale des ensembles limite est indécidable. On va ici prouver le même résultat pour les ensembles  $\mu$ -limite.

### 4.5.1 Propriétés d'ensembles $\mu$ -limite

Pour formaliser les propriétés d'ensembles  $\mu$ -limite, on définit un ensemble infini dénombrable  $X_\alpha = \{\alpha_0, \alpha_1, \dots, \alpha_n, \dots\}$ , et l'alphabet de tout automate cellulaire sera un sous ensemble fini de  $X_\alpha$ .

**Définition 4.5.1.** Une *propriété* d'ensembles  $\mu$ -limite d'automates cellulaires est une famille  $\mathcal{P} \subseteq \mathfrak{P}(X_\alpha^{\mathbb{Z}})$  d'ensembles de configurations. On dit qu'un ensemble  $\mu$ -limite a une  $\mathcal{P}$  propriété s'il fait partie de cette famille.

La  $\mu$ -nilpotence est une propriété d'ensembles  $\mu$ -limite déterminée par la famille  $\{\{\alpha_i^{\mathbb{Z}}\}, i \in \mathbb{N}\}$ . Une propriété des ensembles  $\mu$ -limite est naturellement aussi une propriété des langages  $\mu$ -limite, et par exemple, pour  $i$  fixé, le fait que le langage  $\mu$ -limite contienne  $\alpha_i$  est une propriété.

En revanche, une propriété des automates cellulaires relative à leurs ensembles  $\mu$ -limite n'est pas nécessairement une propriété des ensembles  $\mu$ -limite. Deux automates cellulaires peuvent avoir le même ensemble  $\mu$ -limite sans pour autant avoir le même alphabet. Par exemple la surjectivité est une propriété des automates cellulaires équivalente à avoir un ensemble  $\mu$ -limite plein. Mais si deux automates cellulaires ont pour alphabets  $X_1$  et  $X_2 \subsetneq X_1$ , et pour ensemble  $\mu$ -limite commun  $X_2^{\mathbb{Z}}$ , seul le second est surjectif.

On va donc montrer un théorème de Rice à alphabet non borné. Dans le cas des ensembles limites, P. Guillon et G. Richard [16] ont démontré une variante de ce théorème à alphabet fixé. L'adaptation de la démonstration au cas des ensembles  $\mu$ -limite est en cours.

On dit qu'une propriété est *triviale* lorsque soit tous les ensembles  $\mu$ -limite d'automates cellulaires ont cette propriété, soit aucun ne l'a.

### 4.5.2 Engendrer une configuration générique

On aura besoin dans ce qui suit de calculer des préfixes arbitrairement grands d'une configuration générique pour un alphabet donné. On va utiliser une configuration particulière :

**Définition 4.5.2.** Un mot sur l'alphabet  $X$  est dit *de Bruijn* d'ordre  $n \in \mathbb{N}$  lorsqu'il est de taille  $|X|^n + n - 1$  et que tout mot de taille  $n$  en est un facteur.

Ces mots sont les plus petits ayant tout mot de taille  $n$  comme facteur. Il en existe plusieurs pour chaque  $n$ , et on note  $DB(n)$  le mot de Bruijn particulier construit comme décrit par H. Fredricksen et I. Kessler [14].

**Définition 4.5.3.** On note  $c_{DB} = DB(0)DB(1)DB(2)\dots$  la configuration formée par la concaténation de tous les mots de Bruijn d'ordre  $n \in \mathbb{N}$ . Cette configuration est elle aussi dite *de Bruijn*.

*Remarque 4.5.1.* Cette configuration est une configuration univers, c'est à dire que tout mot fini en est un sous mot.

On va aussi montrer que la densité de chaque mot varie peu en fonction de la taille du préfixe considéré.

**Lemme 4.5.1.** *Il existe deux constantes  $A, B > 0$ , telles que pour tout  $l \geq 1$  et tout  $u \in X^l$ , la densité de  $u$  dans tout préfixe de  $c_{DB}$  de taille  $L \geq |X|^{2l}$  soit  $A|X|^{-|u|} \leq d_{c_{DB}[0, L-1]}(u) \leq B|X|^{-|u|}$ .*

*Démonstration.* On commence par montrer la minoration. On note

$$l_i = \sum_{j < i} |DB(j)| = \frac{|X|^i - 1}{|X| - 1} + \frac{(i-1)(i-2)}{2}$$

la somme des tailles des  $i$  premiers mots de Bruijn. Soit  $u \in X^l, l \in \mathbb{N}$ . On prend un préfixe  $w$  de  $c_{DB}$  de taille  $L \in \mathbb{N}$ , avec  $l_i \leq L < l_{i+1}$  pour un certain  $i$ .

- Pour  $j < l$ , on minore le nombre de  $u$  dans  $DB(j)$  par 0.
- Pour  $l \leq j \leq i-1$  on minore le nombre de  $u$  dans  $DB(j)$  par  $|X|^{j-l}$  (mots de taille  $j$  commençant par  $u$ ).
- Pour  $j = i$ , on minore le nombre de  $u$  dans  $DB(j)$  par 0.

Si  $i \geq l+1$ , on a donc

$$d_w(u) \geq \frac{\sum_{l \leq j < i} |X|^{j-l}}{l_{i+1}} \geq |X|^{-l-1} \frac{1 - |X|^{l-i}}{1 - |X|^{-i-1} + \frac{i(i-1)}{2}(|X| - 1)|X|^{-i-1}}$$

On pose  $A = \frac{1-|X|^{-1}}{1+|X|^{-2}}$  et  $d_w(u) \geq |X|^{-l-1}A$  pour  $i \geq 2l$ .

Pour la majoration, on remarque que le nombre de  $u$  dans  $DB(l+i)$  est inférieur à  $|X|^i + i$  : on peut compléter  $u$  de  $|X|^i$  manières différentes en un mot de  $X^{l+i}$ , et on ajoute les  $i+l-1$  positions où  $u$  peut apparaître mais qui ne sont pas des débuts de sous-mots de  $DB(l+i)$  de taille  $l+i$ . Le même raisonnement permet ensuite de majorer  $d_w(u)$  par  $B|X|^{-|u|}$  pour  $i \geq 2l$ .  $\square$

En particulier,  $c_{DB}$  est générique.

Comme décrit par H. Fredricksen et I. Kessler [14], produire le mot  $DB(n)$  pour  $n \in \mathbb{N}$  nécessite un espace de travail en  $O(n)$  et un temps en  $O(|X|^n)$ .

### 4.5.3 Théorème de Rice

**Théorème 4.5.2.** *Soit  $\mathcal{P}$  une propriété d'ensembles  $\mu$ -limite,  $\mathcal{P}$  est soit triviale soit indécidable.*

On va montrer ce théorème en utilisant une réduction au problème de la  $\mu$ -nilpotence. Plus précisément, on utilisera un automate  $\mathcal{H}$  et, pour tout  $e \in \mathbb{N}$ , on note  $\mathcal{P}_e = \{\{\alpha_e^{\mathbb{Z}}\}\}$ . Décider si  $\Lambda_\mu(\mathcal{H})$  a la propriété  $\mathcal{P}_e$  est plus difficile que décider la  $\mu$ -nilpotence de  $\mathcal{H}$ . En effet les résultats des tests de  $\mathcal{P}_s, s \in X_{\mathcal{H}}$  permettent de décider la  $\mu$ -nilpotence de  $\mathcal{H}$ .

La méthode est semblable à celle utilisée par J. Kari pour les ensembles limite.

La construction que l'on va décrire permettra de prouver la proposition suivante :

**Proposition 4.5.3.** *Soit  $\mathcal{H}$  et  $\mathcal{A}$  des automates cellulaires, pour  $e \in \mathbb{N}$ , on peut construire un automate cellulaire  $\mathcal{B}$  tel que :*

- si  $\Lambda_\mu(\mathcal{H}) \in \mathcal{P}_e$ , alors  $\Lambda_\mu(\mathcal{B}) = \Lambda_\mu(\mathcal{A})$  ;
- sinon  $\Lambda_\mu(\mathcal{B}) = X_{\mathcal{A}}^{\mathbb{Z}}$ .

Après la démonstration de cette proposition, on l'utilisera pour montrer qu'une propriété non triviale  $\mathcal{P}$  est indécidable en prenant deux automates cellulaires  $\mathcal{A}_1$  et  $\mathcal{A}_2$  sur le même alphabet tels que l'ensemble  $\mu$ -limite d'un seul des deux ait la propriété  $\mathcal{P}$ . On obtient alors deux automates cellulaires  $\mathcal{B}_1$  et  $\mathcal{B}_2$  en fonction d'un même  $\mathcal{H}$ . Décider la propriété  $\mathcal{P}$  sur  $\mathcal{B}_1$  et  $\mathcal{B}_2$  reviendrait alors à décider  $\mathcal{P}_e$  sur  $\mathcal{H}$  qui a été démontrée indécidable [4]

*Démonstration.* On décrit la construction. On utilise encore une fois l'automate du lemme 4.3.3 que l'on modifie légèrement : les états  $\boxed{*}$  qui initient des compteurs vont porter un bit d'information supplémentaire. On avait quatre états  $\boxed{*}$ , il y en aura donc huit différents. Le comportement est globalement le même, et le bit supplémentaire est conservé par le segment engendré par l'état  $\boxed{*}$ . Lors des fusions, ces états vont se grouper dans les segments pour former des mots qu'on note  $m_s$  pour

010110100		
010	11	0100

FIGURE 4.10 – Fusion de trois segments et constitution du mot aléatoire du nouveau segment.

le segment  $s$ , et tous les mots des prédécesseurs sont concaténés lors d'une fusion pour former le nouveau mot du segment, comme illustré par la figure 4.10. Ainsi, à tout instant, si l'on est parti d'une configuration normale, ces mots sont répartis selon une distribution équiprobable parmi les segments. De plus pour un segment  $s$  contenu entre les cellules  $a$  et  $b$ ,  $|m_s|$  est égale au nombre d'états  $\boxed{*}$  entre  $a$  et  $b$  sur la configuration initiale :  $(\frac{8}{|\mathcal{X}|}) \times (b - a)$  en moyenne pour une configuration initiale normale. On borne à chaque fusion  $|m_s|$  par  $\sqrt{b - a}$  ainsi  $m_s$  peut être stocké sur l'espace de mémorisation du temps du segment  $s$ .

La figure 4.11 illustre le calcul à l'intérieur d'un segment. Dans le segment  $s$  entre les temps  $t_i$  et  $t_{i+1} - 1$ , on commence par calculer  $|s|$  et  $\nu(i) = \lfloor \log \log i \rfloor$ , puis, comme décrit en 1.3.2, la machine de Turing simule l'automate  $\mathcal{H}$  sur le mot fini  $m_s$  pendant  $\nu(i)$  étapes si  $m_s$  est assez grand. Cela produit un mot  $f(m_s)$  de taille  $|m_s| - 2\nu(i)r_{\mathcal{H}}$ . On distingue alors deux cas :

- Si ce mot appartient à  $\alpha_e^*$  ou si  $m_s$  est trop petit, le segment va simuler  $\mathcal{A}$ . C'est à dire que la machine calcule et écrit dans le segment l'image par  $\mathcal{A}^{\nu(i)}$  d'un préfixe  $w$  de taille  $|s|$  de la configuration  $c_{DB}$  sur l'alphabet  $X_{\mathcal{A}}$ . La machine calcule les états de  $w$  un par un et effectue le calcul de  $\mathcal{A}^{\nu(i)}$  sur tous les facteurs de  $w$  de taille  $2r_{\mathcal{A}}\nu(i) + 1$ . Ainsi, en un seul passage, elle peut écrire l'image du préfixe de  $c_{DB}$ . La partie centrale de taille  $|s| - 2\nu(i)$  sera bien une image du préfixe de  $c_{DB}$ .
- Si ce n'est pas le cas, le segment ne simule pas  $\mathcal{A}$  et se contente d'écrire un préfixe de taille  $|s|$  de la configuration  $c_{DB}$  sur l'alphabet  $X_{\mathcal{A}}$ .

La machine s'arrête soit à la fin de son calcul, soit de manière à avoir effacé son espace de calcul à  $t_{i+1} - 1$ .

On prend une configuration initiale normale  $c$ . On commence par montrer que le calcul satisfait les spécifications requises pour appliquer le lemme 4.3.3.

POINT 1: *Après le temps  $t_i$  pour  $i$  assez grand, dans tout segment de taille inférieure à  $t_{i+1}^{1/4}$ , le calcul a suffisamment d'espace et a le temps de finir.*

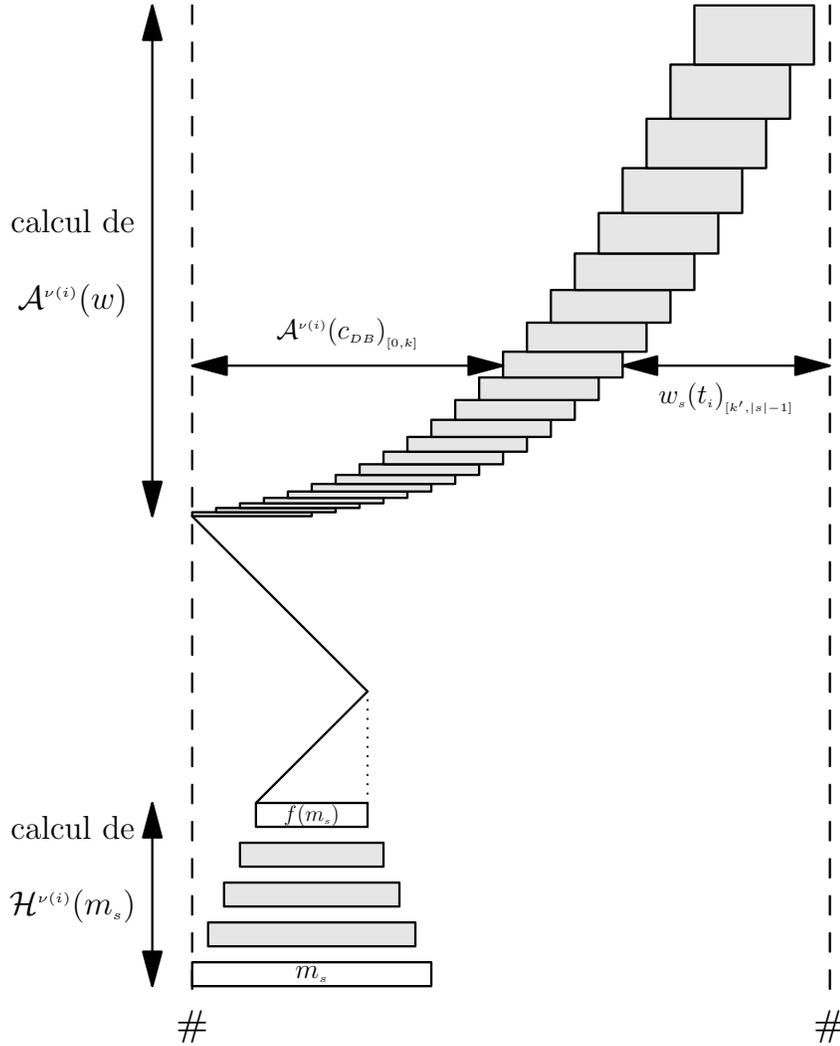


FIGURE 4.11 – Calcul dans un segment, la machine calcule  $f(m_s)$  puis, lorsque  $f(m_s) \in \alpha_e^*$  écrit un préfixe de  $\mathcal{A}^{\nu(i)}(c_{DB})$ . La zone  $\blacksquare$  représente l'espace dédié au calcul de la machine du segment : le temps nécessaire pour produire le préfixe de  $c_{DB}$  est exponentiel est la taille du préfixe.

*Preuve:* En temps  $O(|s|)$ , on peut calculer  $|s|$ , puis constituer le mot  $m_s$ , calculer  $\nu(i)$  et simuler  $\mathcal{H}$  sur le mot  $m_s$ . Ensuite, il faut créer le préfixe de  $c_{DB}$  puis éventuellement simuler  $\mathcal{A}$  sur ce préfixe. La simulation dure  $O(\nu(i)|s|)$  étapes de temps.

Pour calculer le préfixe de  $c_{DB}$  on calcule successivement, les mots  $DB(n)$ ,  $n \leq N$  avec

$$\sum_{n \leq N} |DB(n)| \leq |s|$$

c'est à dire

$$\sum_{n \leq N} |X_{\mathcal{A}}|^n + n - 1 = \frac{|X_{\mathcal{A}}|^{N+1} - 1}{|X_{\mathcal{A}}| - 1} + \frac{N(N-1)}{2} \leq |s|$$

On a donc  $N \leq \log_{|X_{\mathcal{A}}|} |s|$ . Cela assure que la machine a assez d'espace pour calculer les  $DB(n)$  (il suffit de  $O(N)$ ). Et d'autre part, le temps de calcul  $O(|X_{\mathcal{A}}|^n)$  pour  $DB(n)$ , est donc en  $O(|s|)$ .

Finalement, pour  $|s| \leq 2^{(i+1)/4}$ ,  $\nu(i)|s| = o(t_{i+1} - t_i)$ , et il existe  $i_0$  tel que pour tout  $i \geq i_0$  le calcul soit fini dans tout segment de taille inférieure à  $t_{i+1}^{1/4}$ .  $\diamond$  **Point 1**

On montre maintenant qu'il existe une proportion qui ne tend pas vers 0 de segments  $s$  tels que  $m_s$  soit assez grand pour la simulation de  $\mathcal{H}$ .

**POINT 2:** Pour  $i \in \mathbb{N}$ , on note  $M_i = \{s \in S_{t_{i+1}-1}, |m_s| \geq 2 + 2r_{\mathcal{H}}\nu(i)\}$  (où  $r_{\mathcal{H}}$  est le rayon de  $\mathcal{H}$ ) l'ensemble des segments  $s$  acceptables avec  $m_s$  assez grand. Il existe  $i_0 \in \mathbb{N}$ , tel que pour toute cellule  $x \in \mathbb{Z}$  et pour tout  $i \geq i_0$ , dans  $\mathcal{B}^{t_i}(c)$ , on a  $p[x \in \cup_{s \in M_i} s] \geq 1/2$ .

*Preuve:* Soit  $i \in \mathbb{N}$ , et  $s \in S_{t_{i+1}-1}$ . Supposons que  $|m_s| < 2 + 2r_{\mathcal{H}}\nu(i)$ . On regarde la liste de ses prédécesseurs  $s_0, s_1, \dots$ . Lors de la fusion qui a créé  $s$ ,  $m_s$  a été produit comme la concaténation de  $m_{s_0}, m_{s_1}, \dots$

Si  $\sum_j |m_{s_j}| > \sqrt{|s|}$ , il y a eu des pertes d'information, mais dans ce cas

$$|m_s| = \sqrt{|s|} \geq i + 1 \geq 2 + 2r_{\mathcal{H}}\nu(i)$$

Si ce n'est pas le cas  $\sum_j |m_{s_j}| \leq \sqrt{|s|}$ . On continue le processus en regardant les prédécesseurs de chaque  $s_i$ . Pour chacun d'entre eux, s'il y a eu des pertes d'information lors de la fusion,  $|m_{s_j}| = \sqrt{|s_j|}$ , sinon on continue à descendre l'arbre des prédécesseurs. On découpe ainsi le segment  $s$  en un ensemble  $E_0$  de prédécesseurs tels que  $\forall \sigma \in E_0, |m_{\sigma}| = \sqrt{|\sigma|}$  (il y a eu des pertes lors de la création de  $\sigma$ ), et un ensemble  $E_1$  de segments initiaux. On a donc  $|m_s| \geq \sum_{\sigma \in E_0} \sqrt{|\sigma|} + |E_1|$ . Le cas où  $s$  est initial correspond à  $E_0 = \emptyset$  et  $E_1 = \{s\}$ .

On pose  $L = \sum_{\sigma \in E_0} |\sigma|$  et on obtient  $|m_s| \geq \sqrt{L} + |E_1|$ . Si  $L \leq (2 + 2r_{\mathcal{H}\nu}(i))^2$  et  $|E_1| \leq 2 + 2r_{\mathcal{H}\nu}(i)$ , cela signifie que sur les  $|s| - L$  cellules occupées par les segments de  $E_1$ , il n'y avait dans  $c$  que  $2 + 2r_{\mathcal{H}\nu}(i)$  états  $\boxtimes$ . la probabilité d'un tel mot est inférieure à

$$\left( \frac{|S| - L}{2 + 2r_{\mathcal{H}\nu}(i)} \right) \left( \frac{|X| - 8}{|X|} \right)^{|s| - L - 2 - 2r_{\mathcal{H}\nu}(i)}$$

et il existe  $i_0 \geq i_1$  tel que pour  $i \geq i_0$ , cette probabilité soit inférieure à  $1/2$ .  $\diamond$  **POINT 2**

**POINT 3:** On note  $Sim_{i,l} = \{s \in S_{t_i}, |s| = l, s \text{ simule } \mathcal{A}\}$ . Pour  $s \in S_{t_i}$ , si  $\Lambda_\mu(\mathcal{H})$  a la propriété  $\mathcal{P}_e$ , alors  $P_{s \in S_{t_i}}[s \in Sim_{i,|s|}] \rightarrow_{i \rightarrow \infty} 1$ . Sinon  $P_{s \in S_{t_i}}[s \notin Sim_{i,|s|}] \not\rightarrow_{i \rightarrow \infty} 0$

*Preuve:* On montre ici que la probabilité pour un segment acceptable de simuler  $\mathcal{A}$  tend vers 1 au cours du temps lorsque  $\Lambda_\mu(\mathcal{H})$  a la propriété  $\mathcal{P}_e$ , et ne tend pas vers 0 sinon. Pour cela, on utilise le fait que le mot  $m_s$  est souvent assez grand dans les segments.

Dans un segment  $s \in S_{t_i}$ , si la taille de  $m_s$  est trop petite, le segment simule  $\mathcal{A}$  donc  $s \in Sim_{i,|s|}$ . Si la taille de  $m_s$  est suffisante, le calcul de  $\mathcal{H}$  se fait sur  $m_s$ . Si  $\Lambda_\mu(\mathcal{H}) \in \mathcal{P}_e$ ,

$$\mu \left( \mathcal{H}^{-\nu(i)}([\alpha_e^*]) \right) \rightarrow_{i \rightarrow \infty} 1$$

Donc

$$P[f(m_s) \in \alpha_e^*] \rightarrow_{i \rightarrow \infty} 1$$

Il s'ensuit que  $P_{s \in S_{t_i}}[s \in Sim_{i,|s|}] \rightarrow_{i \rightarrow \infty} 1$ .

Si  $\Lambda_\mu(\mathcal{H}) \notin \mathcal{P}_e$ , il existe  $\varepsilon > 0$  et une suite  $(i_j)_j$  telle que pour  $j \in \mathbb{N}$  et  $s \in S_{t_{i_j}}$ , si  $m_s$  est assez grand  $P[f(m_s) \notin \alpha_e^*] \geq \varepsilon$ . Or d'après le point 2,  $m_s$  est assez grand avec probabilité au moins  $1/2$ . Donc finalement,  $P_{s \in S_{t_i}}[s \notin Sim_{i,|s|}] \not\rightarrow_{i \rightarrow \infty} 0$ .  $\diamond$  **POINT 3**

On va prouver les points suivants à l'aide du lemme 4.3.3.

**POINT 4:** Si  $\Lambda_\mu(\mathcal{H})$  n'a pas la propriété  $\mathcal{P}_e$ , alors  $\Lambda_\mu(\mathcal{B}) = X_{\mathcal{A}}^{\mathbb{Z}}$ .

*Preuve:* Lorsque  $\Lambda_\mu(\mathcal{H})$  n'a pas la propriété  $\mathcal{P}_e$ , la proportion de segments qui ne simulent pas  $\mathcal{A}$  ne tend pas vers 0. Dans ces segments, la machine écrit un préfixe de la configuration générique, donc tous les mots y apparaissent en densité non négligeable.

Pour tout  $\varepsilon > 0$ , il existe  $i_0$  tel que pour tout  $i \geq i_0$ , et dans tout segment  $s \in S_{t_i}$ ,  $\frac{\log |s| + \sqrt{|s|}}{|s|} \leq \varepsilon$ . Ainsi pour tout mot  $u \notin X_{\mathcal{A}}^{\mathbb{Z}}$ ,  $d_{w_{t_{i+1}-1}(s)}(u) < \varepsilon$  et donc  $u \notin L_\mu(\mathcal{B})$ .

Soit  $u \in X_{\mathcal{A}}^*$ , d'après le lemme 4.5.1, pour tout  $L \geq L_0 = |X|^{2|u|}$ ,  $A|X|^{-|u|} \leq d_{c_{DB}[0, L-1]}(u)$ . D'après le point 3, il existe  $(i_j)_{j \in \mathbb{N}}$  et  $\varepsilon > 0$  tels que pour tout  $l \in \mathbb{N}$ , la probabilité pour qu'un segment  $s \in S_{t_{i_j}}$  avec  $|s| = l$  ne simule pas  $\mathcal{A}$  soit supérieure à  $\varepsilon$ .

Pour tout  $j \in \mathbb{N}$  tel que  $(i_j + 1)^2 \geq L_0$ , dans un segment  $s \in S_{t_{i_j}}$  qui ne simule pas  $\mathcal{A}$ , on a, en tenant compte des cellules allouées au décompte du temps,  $\frac{1}{2}A|X|^{-|u|} \leq d_{wt_{i_j+1-1}(s)}(u)$ . Le cas 1 du lemme 4.3.3 permet de conclure que  $u \in L_\mu(\mathcal{B})$ .  $\diamond$  **Point 4**

**POINT 5:** Si  $\Lambda_\mu(\mathcal{H})$  a la propriété  $\mathcal{P}_e$ , alors  $\Lambda_\mu(\mathcal{A}) \subseteq \Lambda_\mu(\mathcal{B})$ .

*Preuve:*

Lorsque l'ensemble  $\mu$ -limite de  $\mathcal{H}$  est égal à  $\{\alpha_e^{\mathbb{Z}}\}$ , les segments tendent à simuler l'automate  $\mathcal{A}$ , donc ils contiennent des images de préfixes de plus en plus grands de  $c_{DB}$ . Comme  $c_{DB}$  est générique, ses images contiennent les mots du langage  $\mu$ -limite de  $\mathcal{A}$ .

Soit  $u \in L_\mu(\mathcal{A})$ . D'après le point 3, il existe  $i_0 \in \mathbb{N}$  tels que  $\forall i \geq i_0, \forall l \in \mathbb{N}, \forall s \in S_{t_i}$  avec  $|s| = l$ ,  $s$  simule  $\mathcal{A}$  avec probabilité supérieure à  $1/2$ . De plus, il existe  $\varepsilon > 0$  et  $(\tau_n)_n$  tels que

$$\forall n \in \mathbb{N}, \mu(\mathcal{A}^{-\tau_n}([u])) > \varepsilon$$

donc  $\frac{|\text{Ant}^{\tau_n}(u)|}{|X|^{|u|+2\tau_n}} > \varepsilon$ .

Pour tout  $n \in \mathbb{N}, \exists i_n$  tel que  $\nu(i_n) = \tau_n$ . On prend  $n_0 \in \mathbb{N}$  tel que  $\forall n \geq n_0, (i_n + 1)^2 \geq |X_{\mathcal{A}}|^{|u|+2\nu(i_n)}$ .

Soit  $n \geq n_0$ . Soit  $s \in S_{t_{i_n}}$  de taille  $l \in [(i_n + 1)^2, 2^{i_n/4}]$ ,  $s$  contient donc une image d'un préfixe de  $c_{DB}$  avec probabilité  $1/2$  au moins. Puisque l'on a choisi  $i_n$  tel que  $(i_n + 1)^2 \geq |X_{\mathcal{A}}|^{|u|+4\nu(i_n)}$ , le lemme 4.5.1 nous assure que  $\forall v \in \text{Ant}^{\tau_n}(u), d_{c_{DB}[0,|s|-1]}(v) \geq A|X|^{-|v|}$ .

Or

$$d_{wt_{i_n+1-1}(s)}(u) \geq d_{c_{DB}[0,|s|-1]}(\text{Ant}^{\tau_n}(u)) - o(1)$$

en enlevant les état dus au décompte du temps. Et comme  $|\text{Ant}^{\tau_n}(u)| > \varepsilon|X|^{|v|}$ ,

$$d_{wt_{i_n+1-1}(s)}(u) \geq \varepsilon|X|^{|v|}A|X|^{-|v|} - o(1)$$

On peut donc appliquer le cas 1 du lemme 4.3.3 et finalement  $u \in L_\mu(\mathcal{B})$ .  $\diamond$  **Point 5**

**POINT 6:** Si  $\Lambda_\mu(\mathcal{H})$  a la propriété  $\mathcal{P}_e$ , alors  $\Lambda_\mu(\mathcal{B}) \subseteq \Lambda_\mu(\mathcal{A})$ .

*Preuve:* On va maintenant montrer que tout mot qui n'est pas dans le langage  $\mu$ -limite de  $\mathcal{A}$  n'est pas non plus dans le langage  $\mu$ -limite de  $\mathcal{B}$ . Pour cela, comme les segments simulent presque tous l'automate  $\mathcal{A}$ , on montre que seuls les mots qui sont dans les images de  $c_{DB}$  apparaissent fréquemment dans les images d'une configuration normale par  $\mathcal{B}$ .

Soit  $u \notin L_\mu(\mathcal{A})$  et  $\varepsilon > 0$ , il existe  $j_0 \in \mathbb{N}$  tel que  $\forall j \geq j_0, \mu(\mathcal{A}^{-j}([u])) < \varepsilon$ . On pose  $i_0 \in \mathbb{N}$  tel que  $\forall i \geq i_0, \nu(i) \geq j_0$ .

D'après le point 3, il existe  $i_1 \geq i_0$  tel que  $\forall i \geq i_1, \forall s \in S_{t_i}$  de taille  $l \in \mathbb{N}$ ,  $s$  simule  $\mathcal{A}$  avec probabilité au moins  $(1 - \varepsilon)$ .

Soit  $i_2 \geq i_1$  tel que  $\forall i \geq i_2, (i + 1)^2 \geq 2|u| + 4\nu(i)$ . Soit  $i \geq i_2$  et  $s \in S_{t_i}$ . D'après le lemme 4.5.1, il existe  $B \in \mathbb{R}$  tel que  $d_{c_{DB}[0,|s|-1]}(\text{Ant}^{\nu(i)}(u)) \leq B|X|^{-|u|-2\nu(i)}$ .

Avec probabilité  $(1 - \varepsilon)$  au moins, le segment  $s$  simule  $\mathcal{A}$ , donc  $w_{t_{i+1}-1}(s)$  est l'image de  $c_{DB}[0,|s|-1]$  par  $\mathcal{A}^{\nu(i)}$  modulo les états dédiés à l'âge de la construction. On a donc

$$d_{w_{t_{i+1}-1}(s)}(u) \leq d_{c_{DB}[0,|s|-1]}(\text{Ant}^{\nu(i)}(u))$$

et finalement  $d_{w_{t_{i+1}-1}(s)}(u) \leq B\varepsilon$ . Ce qui permet d'appliquer le cas 2 du lemme 4.3.3 pour conclure. ◇ Point 6

Cela conclut la démonstration de la proposition. □

On va maintenant utiliser la proposition précédente pour prouver le théorème 4.5.2. Pour cela, pour une propriété  $\mathcal{P}$  des ensembles  $\mu$ -limite, on prend deux automates cellulaires tels que l'ensemble  $\mu$ -limite de l'un d'entre eux exactement a la propriété  $\mathcal{P}$ . On commence par s'assurer que l'on peut prendre ces deux automates cellulaires sur le même alphabet.

**Lemme 4.5.4.** *Soit  $\mathcal{P}$  une propriété non triviale des ensembles  $\mu$ -limite, il existe des automates cellulaires  $\mathcal{A}_1$  et  $\mathcal{A}_2$  sur le même alphabet  $X$  tels qu'un exactement de leurs ensembles  $\mu$ -limite ait la propriété  $\mathcal{P}$ .*

*Démonstration.* Comme  $\mathcal{P}$  n'est pas triviale, il existe  $\mathcal{D}_0$  et  $\mathcal{D}_1$  deux automates cellulaires sur les alphabets  $X_0$  et  $X_1$  tels qu'un seul parmi  $\Lambda_\mu(\mathcal{D}_0)$  et  $\Lambda_\mu(\mathcal{D}_1)$  ait la propriété  $\mathcal{P}$ . On va remplacer les états de chaque automate par plusieurs copies de chacun pour augmenter les alphabets, et ainsi leur donner le même cardinal. On fait ensuite disparaître les copies lors de la première étape de calcul.

On pose  $c = |X_0|$  et  $d = |X_1|$ , puis on note  $e = c \times d$  et on prend un alphabet de taille  $e : X = \{\alpha_{i_0}, \alpha_{i_1}, \dots, \alpha_{i_e}\}$ , tel que  $X_0 \cup X_1 \subseteq X$ . On partitionne  $X$  deux fois :

- en  $c$  ensembles  $\{S_0(\gamma), \gamma \in X_0\}$  avec pour tous  $\gamma, \gamma' \in X_0$ ,  $\gamma \in S_0(\gamma)$  et  $|S_0(\gamma)| = |S_0(\gamma')| = |X_1|$ ;
- en  $d$  ensembles  $\{S_1(\gamma), \gamma \in X_1\}$  avec pour tous  $\gamma, \gamma' \in X_1$ ,  $\gamma \in S_1(\gamma)$  et  $|S_1(\gamma)| = |S_1(\gamma')| = |X_0|$ .

On définit maintenant les règles locales des automates  $\mathcal{A}_0$  et  $\mathcal{A}_1$  sur l'alphabet  $X$  :

- $\delta_{\mathcal{A}_0}(\alpha_x, \alpha_y, \alpha_z) = \alpha_t$  lorsque  $\delta_{\mathcal{D}_0}(\alpha_{x'}, \alpha_{y'}, \alpha_{z'}) = \alpha_t$  où  $\alpha_x \in S_0(\alpha_{x'})$ ,  $\alpha_y \in S_0(\alpha_{y'})$  et  $\alpha_z \in S_0(\alpha_{z'})$ ;
- $\delta_{\mathcal{A}_1}(\alpha_x, \alpha_y, \alpha_z) = \alpha_t$  lorsque  $\delta_{\mathcal{D}_1}(\alpha_{x'}, \alpha_{y'}, \alpha_{z'}) = \alpha_t$  où  $\alpha_x \in S_1(\alpha_{x'})$ ,  $\alpha_y \in S_1(\alpha_{y'})$  et  $\alpha_z \in S_1(\alpha_{z'})$ .

Clairement, la première image d'une configuration normale sur  $X$  par  $\mathcal{A}_i$  est équivalente à l'image d'une configuration normale sur  $X_i$  par  $\mathcal{D}_i$ . Donc les ensembles  $\mu$ -limite sont les mêmes.  $\square$

On peut maintenant conclure la démonstration du théorème 4.5.2 :

*Démonstration.* Soit  $\mathcal{P}$  une propriété non triviale, d'après le lemme précédent, il existe deux automates cellulaires  $\mathcal{A}_1$  et  $\mathcal{A}_2$  sur l'alphabet  $X$ , tels qu'un seul de leurs ensembles  $\mu$ -limite ait la propriété. On utilise alors deux fois la proposition 4.5.3 pour obtenir les automates  $\mathcal{B}_0$  et  $\mathcal{B}_1$  à partir de  $\mathcal{A}_1$  et  $\mathcal{A}_2$  et d'un automate  $\mathcal{H}$  donné.

Supposons que l'on puisse décider si l'ensemble  $\mu$ -limite d'un automate cellulaire a la propriété  $\mathcal{P}$ . Alors on utilise l'algorithme de décision sur  $\mathcal{B}_0$  et  $\mathcal{B}_1$ . Si les réponses sont différentes, alors  $\Lambda_\mu(\mathcal{B}_0) \neq \Lambda_\mu(\mathcal{B}_1)$  donc  $\Lambda_\mu(\mathcal{H})$  a la propriété  $\mathcal{P}_e$ . En revanche, pour que les réponses soient identiques, il faut  $\Lambda_\mu(\mathcal{B}_0) = X^{\mathbb{Z}} = \Lambda_\mu(\mathcal{B}_1)$  et  $\Lambda_\mu(\mathcal{H})$  n'a pas la propriété  $\mathcal{P}_e$ .

On a donc un algorithme pour décider  $\mathcal{P}_e$ , ce qui contredit l'indécidabilité de la  $\mu$ -nilpotence [4].  $\square$

*Remarque 4.5.2.* On a ici montré que toute propriété est au moins aussi indécidable que la  $\mu$ -nilpotence. Et, au contraire de la nilpotence qui est récursivement énumérable (parce que la configuration uniforme est atteinte en temps fini [7]), la position de la  $\mu$ -nilpotence dans la hiérarchie n'est pas encore déterminée. Un résultat sur cette question, entraînerait donc un résultat pour toute propriété non triviale. Toutefois, la réduction que l'on a employé est une réduction Turing et même *truth-table* mais pas *many-one*, on ne peut donc pas transposer un éventuel résultat tel quel.

## 4.6 Moyenne de Cesaro

Dans cette partie, on va montrer que la définition d'ensemble  $\mu$ -limite telle qu'on l'a choisie n'est pas la seule définition envisageable. Un mot n'a besoin, pour être dans le langage  $\mu$ -limite d'un automate cellulaire, que d'apparaître beaucoup lors de certaines étapes. Mais ces étapes peuvent être très rares. On illustrera cette particularité par un exemple, puis on proposera une définition différente et enfin, on donnera quelques points de comparaison.

### 4.6.1 Exemple

On considère une fois de plus un automate cellulaire  $\mathcal{A}$  avec la construction compteurs définie en section 4.3. Le calcul dans un segment  $s$  au temps  $t_i$  consiste à :

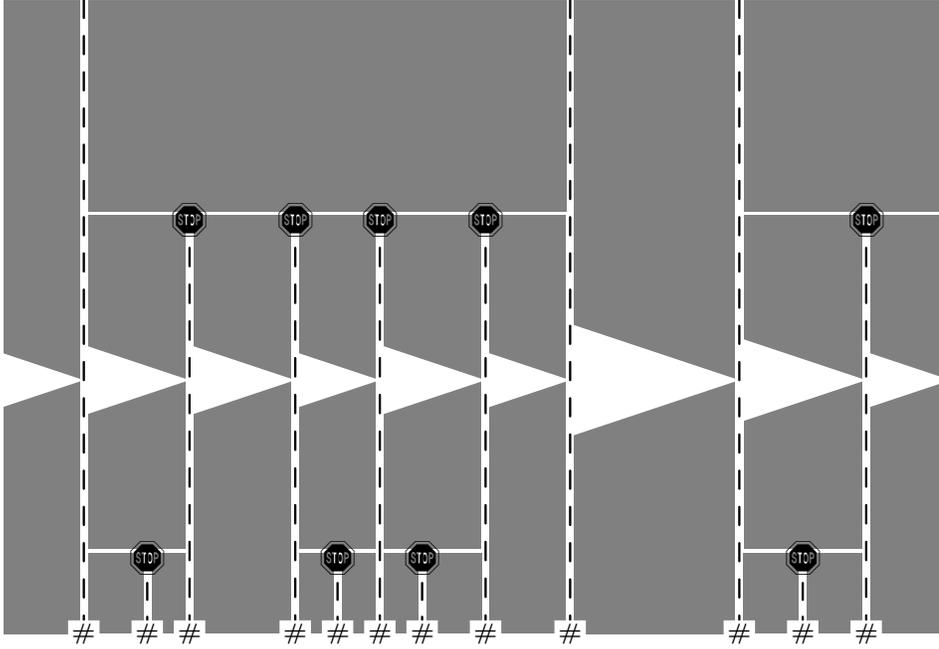


FIGURE 4.12 – Tous les segments contiennent un mot de  $\blacksquare^*$  sauf autour du temps  $\frac{t_i+t_{i+1}}{2}$  où ils contiennent un mot de  $\square^*$ .

- mesurer la taille du segment ;
- écrire l'état  $\blacksquare$  sur tout le segment ;
- écrire l'état  $\square$  sur tout le segment à partir de l'étape  $2^i + 2^{i-1} - |s|$  jusqu'à  $2^i + 2^{i-1}$  où tout le segment contient  $\square$ , puis réécrire l'état  $\blacksquare$  partout, ce qui se finit en  $2^i + 2^{i-1} + |s|$ .

$\mathcal{A}$  est illustré par la figure 4.12.

On voit rapidement que  $\Lambda_\mu(\mathcal{A}) = \{\blacksquare^{\mathbb{Z}}, \square^{\mathbb{Z}}\}$  puisque tous les autres mots ont une densité faible dans les segments. Comme les segments sont synchronisés, la configuration est quasiment entièrement remplie de  $\square$  au temps  $(t_i + t_{i-1})/2$ . Pourtant, les deux états ne jouent pas le même rôle :  $\square$  n'est présent qu'aux temps proches de  $(t_i + t_{i-1})/2$ . En particulier, une moyenne temporelle le ferait disparaître. On va donc introduire une nouvelle définition qui ajoute une moyenne temporelle.

### 4.6.2 Définition

On change la condition pour appartenir au langage  $\mu$ -limite, au lieu d'une limite supérieure de la mesure des antécédents strictement positive, on va utiliser une moyenne de Cesaro.

**Définition 4.6.1.** Un mot  $u \in X^*$  est dans le langage  $\mu$ -moyen  $M_\mu(\mathcal{A})$  de  $\mathcal{A}$  lorsque

$$\frac{1}{n+1} \sum_{i \leq n} \mu(\mathcal{A}^{-i}([u])) \xrightarrow{n \rightarrow \infty} 0$$

Et l'ensemble  $\mu$ -moyen de  $\mathcal{A}$  est défini par

$$\Gamma_\mu(\mathcal{A}) = \{c \in X^\mathbb{Z}, \forall i, j \in \mathbb{Z}, c_{[i,j]} \in M_\mu(\mathcal{A})\}$$

L'ensemble  $\mu$ -moyen de l'automate décrit dans la partie 4.6.1 est donc  $\Gamma_\mu(\mathcal{A}) = \{\square^\mathbb{Z}\}$ .

On peut transférer les résultats vus sur les ensembles  $\mu$ -limite aux ensembles  $\mu$ -moyen, par exemple pour les sous-shifts atteignables ou pour le théorème de Rice. On va voir en revanche que les définitions ne coïncident pas sur des propriétés simples.

### 4.6.3 Presque nilpotence

On peut définir pour les ensembles moyens une notion de *presque nilpotence* calquée sur la  $\mu$ -nilpotence pour les ensembles  $\mu$ -limite. Si on considère la définition restrictive de  $\mu$ -nilpotence que l'on a utilisée jusqu'ici (une seule configuration dans l'ensemble  $\mu$ -limite), la  $\mu$ -nilpotence implique la presque nilpotence. En revanche, l'inverse est infirmé par l'exemple précédent dans lequel l'ensemble  $\mu$ -moyen est  $\Gamma_\mu(\mathcal{A}) = \{\square^\mathbb{Z}\}$  alors que  $\Lambda_\mu(\mathcal{A}) = \{\square^\mathbb{Z}, \square^\mathbb{Z}\}$ .

Si on relâche la définition en acceptant comme  $\mu$ -nilpotents les automates cellulaires dont l'ensemble  $\mu$ -limite est fini, on va montrer que la  $\mu$ -nilpotence n'implique toujours pas la presque nilpotence. En effet, on modifie l'automate  $\mathcal{A}$  de l'exemple pour obtenir un automate tel que  $\Lambda_\mu(\mathcal{A}) = \{(\square\square^k)^\mathbb{Z}, k \in \mathbb{N}\} \cup \{\square^\mathbb{Z}\}$  alors que  $\Gamma_\mu(\mathcal{A}) = \{\square^\mathbb{Z}\}$ .

Pour cela, le calcul dans les segments à  $t_i$  consiste à écrire un mot de  $(\square\square)^*$  dans le segment, puis le remplacer par un mot de  $(\square\square^2)^*$ , puis remplacer celui-ci par un mot de  $(\square\square^3)^*$  et ainsi de suite, jusqu'à  $t_{i+1} - 1$ . Ainsi  $\square$  n'apparaît souvent qu'au début du calcul dans un segment. Il disparaît donc de l'ensemble  $\mu$ -moyen.

On peut donc obtenir un ensemble  $\mu$ -limite infini en même temps qu'un ensemble  $\mu$ -moyen réduit à une configuration.

# Bilan et perspectives

Nous avons commencé par étudier les ensembles de conséquences de mots finis, qui sont un outil adapté à l'étude de la dynamique directionnelle et en particulier l'étude des mots bloquants. Nous avons montré que l'on pouvait obtenir une large classe d'ensembles de conséquences, avec des frontières non périodiques. La calculabilité des frontières est la seule contrainte que l'on a sur la forme d'un ensemble de conséquences. Chercher une caractérisation de l'ensemble des ensembles de conséquences réalisables passe donc par trouver un exemple de fonction calculable non réalisable comme frontière, ou montrer qu'elles le sont toutes.

Pour les ensembles de conséquences « pleins », c'est à dire ceux pour lesquels tous les sites entre les frontières sont dans les conséquences, l'étude des frontières réalisables permet d'étudier les directions bloquantes. Mais les conséquences ont aussi un intérêt propre. On peut s'intéresser aux mots ayant un ensemble de conséquences épars, donc un mot qui sans être bloquant « filtre » l'information. Comment la communication s'opère-t-elle dans un automate cellulaire possédant un tel mot ? L'existence d'un tel ensemble de conséquences large mais non bloquant et ayant une structure complexe serait intéressant à ce titre.

Nous avons ensuite vu les mots ayant des ensembles de conséquences larges et pleins comme des mots bloquants selon des directions diverses. Nous avons généralisé la caractérisation de M. Sablik [37] concernant les ensembles de directions bloquantes d'un automate cellulaire. Dans le cas de la restriction aux droites, nous avons obtenu une caractérisation de ces ensembles, mais le cas général reste ouvert. Les bornes de cet ensemble sont calculables et une caractérisation de ces ensembles revient à donner une caractérisation des ensembles de conséquences pleins.

L'équivalence entre existence d'un point d'équicontinuité et d'un mot bloquant est un résultat marquant de dynamique directionnelle, mais il est remis en cause pour les fonctions à variations non bornées. En effet, dans ce cas, savoir si l'existence d'un point d'équicontinuité entraîne l'existence d'un mot bloquant selon la même direction reste un problème ouvert.

La généralisation de la notion d'expansivité à des directions quelconques n'est,

elle, pas évidente. Nous avons vu en particulier que l'on pouvait rendre l'identité expansive pour certaines fonctions. Ainsi, il est peut être préférable de se restreindre à des sous ensembles de fonctions comme les fonctions monotones.

La dernière partie de ce mémoire est consacrée aux ensembles  $\mu$ -limites et notamment aux questions de calculabilité sur ces ensembles. Dans le cas où il existe un mot bloquant, nous avons généralisé à des directions quelconques un résultat [4] affirmant que le langage  $\mu$ -limite est récursivement énumérable. Dans le cas général, nous avons montré que l'appartenance à ce langage pouvait être un problème  $\Sigma_2$ -complet. La seule borne supérieure à cette complexité étant  $\Sigma_3$ , il reste un pas à faire pour connaître l'exact éventail de complexités atteignables.

Une autre question de calculabilité concerne les propriétés d'ensembles  $\mu$ -limites. Nous avons montré que toutes les propriétés non triviales étaient indécidables. Obtenir la complexité exacte de la  $\mu$ -nilpotence permettrait de donner une borne inférieure plus précise à la complexité d'une propriété non triviale. Il n'existe pour le moment pas de borne supérieure, donc il est également intéressant de chercher des propriétés ayant une très haute complexité. La complexité de certaines propriétés naturelles comme le fait d'être un sous-shift de type fini ou sofique n'est pas connue non plus à ce jour. On peut encore faire un parallèle supplémentaire avec l'ensemble limite, en s'intéressant à un théorème de Rice avec nombre d'états fixé. P. Guillon et G. Richard [16] ont montré qu'une seule propriété non triviale est décidable en bornant le nombre d'états de l'automate : la surjectivité. Il reste à étudier la transposition de ce résultat aux ensembles  $\mu$ -limites, ce travail est en cours et la méthode de preuve de P. Guillon et G. Richard combinée à la construction présentée dans ce document devrait aboutir à une démonstration du résultat.

Nous avons commencé à décrire l'ensemble des ensembles  $\mu$ -limite atteignables en donnant l'ensemble des sous-shifts soifiques atteignables. Les contraintes à respecter pour être un ensemble  $\mu$ -limite sont en revanche encore peu connues. Une caractérisation des sous-shifts atteignables peut éventuellement être obtenue en combinant des contraintes de calculabilité et une restriction de type « non-errant ».

La question de la mesure considérée se pose aussi. Nous n'avons regardé dans tout ce mémoire que la mesure uniforme, et les résultats sont *a priori* généralisables à d'autres mesures. Le cas des mesures de Bernoulli vient naturellement, mais on peut aussi étendre certains résultats à des classes plus larges de mesures, en particulier les mesures à support total ou les mesures ergodiques.

Enfin, la construction compteurs est un élément important de ce rapport, elle est utilisée à la fois pour construire des mots bloquants selon diverses directions et pour construire des ensembles  $\mu$ -limites particuliers. Elle pourrait être utile dans d'autres circonstances et il peut aussi être intéressant de la généraliser aux dimensions

supérieures. Pour la dimension 2 par exemple, une méthode pourrait consister en des compteurs faisant croître des zones rectangulaires jusqu'à rencontrer une zone similaire. On recouvrirait alors le plan par des zones de calcul rectangulaires et des zones libres qui seraient assimilées lors des fusions.



# Bibliographie

- [1] Klaus Ambros-Spies, Klaus Weihrauch et Xizhong Zheng. Weakly computable real numbers. *Journal of Complexity*, 16 :676–690, Décembre 2000.
- [2] François Blanchard et Alejandro Maass. Dynamical properties of expansive one-sided cellular automata. *Israel Journal of Mathematics*, 99 :149–174, 1997.
- [3] Laurent Boyer, Martin Delacourt et Mathieu Sablik. Construction of  $\mu$ -limit sets. Dans *Proceedings of JAC 2010*, pages 76–87, Turku, Finland, Décembre 2010.
- [4] Laurent Boyer, Victor Poupet et Guillaume Theyssier. On the complexity of limit sets of cellular automata associated with probability measures. Dans *MFCSS*, volume 4162 de *Lecture Notes in Computer Science*, pages 190–201. Springer, 2006.
- [5] B. Chopard et M. Droz. *Cellular automata modeling of physical systems*. Cambridge University Press, Mai 1998.
- [6] Alonzo Church. An unsolvable problem of elementary number theory. *The American Journal Of Mathematics*, 58 :345–363, 1936.
- [7] Karel Culik, II, Jan K. Pachl et Sheng Yu. On the limit sets of cellular automata. *SIAM Journal on Computing*, 18 :831–842, Août 1989.
- [8] Karel Culik, II et Sheng Yu. Undecidability of ca classification schemes. *Complex Systems*, 2 :177–190, Avril 1988.
- [9] Martin Delacourt. Rice’s theorem for  $\mu$ -limit sets of cellular automata. Dans *ICALP (2)*, volume 6756 de *Lecture Notes in Computer Science*, pages 89–100. Springer, 2011.
- [10] Martin Delacourt, Victor Poupet, Mathieu Sablik et Guillaume Theyssier. Directional dynamics along arbitrary curves in cellular automata. *Theoretical Computer Science*, 412 :3800–3821, Juillet 2011.
- [11] Patrick C. Fischer. Generation of primes by a one-dimensional real-time iterative array. *Journal of the ACM*, 12 :388–394, Juillet 1965.

- [12] Philippe Flajolet et Robert Sedgewick. *Analytic combinatorics*. Cambridge University Press, Janvier 2009.
- [13] Pytheas Fogg. *Substitutions in dynamics, arithmetics, and combinatorics*. Springer, 2002.
- [14] Harold Fredricksen et Irving Kessler. Lexicographic compositions and de Bruijn sequences. *Journal of Combinatorial Theory, Series A*, 22(1) :17 – 30, 1977.
- [15] Martin Gardner. The fantastic combinations of John Conway’s new solitaire game “life”. *Scientific American*, 223 :120–123, Octobre 1970.
- [16] Pierre Guillon et Gaétan Richard. Revisiting the rice theorem of cellular automata. Dans *STACS’10*, pages 441–452, 2010.
- [17] Gustav Arnold Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Theory of Computing Systems*, 3(4) :320–375, Décembre 1969.
- [18] Mike Hurley. Attractors in cellular automata. *Ergodic Theory and Dynamical Systems*, 10(01) :131–140, 1990.
- [19] Mike Hurley. Ergodic aspects of cellular automata. *Ergodic Theory and Dynamical Systems*, 10(04) :671–685, 1990.
- [20] Jarkko Kari. Rice’s theorem for the limit sets of cellular automata. *Theoretical Computer Science*, 127 :229–254, Mai 1994.
- [21] Jarko Kari. The nilpotency problem of one-dimensional cellular automata. *SIAM J. Comput.*, 21 :571–586, Juin 1992.
- [22] Petr Kůrka. Languages, equicontinuity and attractors in cellular automata. *Ergodic Theory and Dynamical Systems*, 17(02) :417–433, 1997.
- [23] Petr Kůrka et Alejandro Maass. Limit sets of cellular automata associated to probability measures. *Journal of Statistical Physics*, 100 :1031–1047, 2000. 10.1023/A :1018706923831.
- [24] Douglas A. Lind et Brian H. Marcus. *An introduction to symbolic dynamics and coding*. Cambridge, 1995.
- [25] Jacques Mazoyer. A six-state minimal time solution to the firing squad synchronization problem. *Theoretical Computer Science*, 50 :183–238, Janvier 1987.
- [26] Jacques Mazoyer et Véronique Terrier. Signals in one-dimensional cellular automata. *Theoretical Computer Science*, 217(1) :53 – 80, 1999.
- [27] Edward Forrest Moore. Machine models of self-reproduction. Dans *Mathematical problems in biological sciences (Proceedings of Symposia in applied mathematics)*. American Mathematical Society, 1962.
- [28] John Myhill. Shorter note : The converse of Moore’s Garden-of-Eden theorem. *Proceedings of the American Mathematical Society*, 14(4) :685–686, 1963.

- [29] John von Neumann. *Theory of self-reproducing automata*. University of Illinois Press, 1966.
- [30] Piergiorgio Odifreddi. Chapter x recursively enumerable degrees. Dans *Classical Recursion Theory*, volume 143 de *Studies in Logic and the Foundations of Mathematics*, pages 455 – 639. Elsevier, 1999.
- [31] Nicolas Ollinger. The quest for small universal cellular automata. Dans *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, ICALP '02, pages 318–329. Springer-Verlag, 2002.
- [32] Nicolas Ollinger et Gaétan Richard. Collisions and their catenations : ultimately periodic tilings of the plane. Dans *Fifth Ifip International Conference On Theoretical Computer Science , TCS 2008*, volume 273 de *IFIP International Federation for Information Processing*, pages 229–240. Springer Boston, 2008.
- [33] Nicolas Ollinger et Gaétan Richard. Automata on the plane vs particles and collisions. *Theoretical Computer Science*, 410(27-29) :2767 – 2773, 2009.
- [34] Marcus Pivato. Algebraic invariants for crystallographic defects in cellular automata. *Ergodic Theory and Dynamical Systems*, 27(01) :199–240, 2007.
- [35] Marcus Pivato. Defect particle kinematics in one-dimensional cellular automata. *Theoretical Computer Science*, 377(1-3) :205 – 228, 2007.
- [36] Marcus Pivato. Spectral domain boundaries in cellular automata. *Fundamenta Informaticae*, 78 :417–447, Août 2007.
- [37] Mathieu Sablik. Directional dynamics for cellular automata : A sensitivity to initial condition approach. *Theoretical Computer Science*, 400(1-3) :1 – 18, 2008.
- [38] Alvy Ray Smith, III. Simple computation-universal cellular spaces. *Journal of the ACM*, 18 :339–353, Juillet 1971.
- [39] Alan Mathison Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(1) :230–265, Janvier 1937.
- [40] Abraham Waksman. An optimum solution to the firing squad synchronization problem. *Information and Control*, 9(1) :66 – 78, 1966.
- [41] Stephen Wolfram. Universality and complexity in cellular automata. *Physica D : Nonlinear Phenomena*, 10(1-2) :1–35, Janvier 1984.



# Index

- $\mu$ -limite, ensemble, 74
  - propriété, 105
- $\mu$ -nilpotence, 80
- antécédents d'un mot, 74
- antécédents d'un site, 6
- automate cellulaire, 4
  - équicontinu, 11, 47
    - direction, 47
  - bloquant, 46
    - direction, 46
  - expansif, 12, 49
    - direction, 49
  - MAX, 8
  - réversible, 5
  - sensible, 48
    - direction, 48
- compteurs, 33
- configuration, 1
  - générique, 78
  - normale, 78
- conséquences, ensemble de, 20
- cylindre, 2
- décalage, 2
- densité, 77
- diagramme espace-temps, 6
- ensemble limite, 8
- état
  - envahissant, 8
  - persistant, 8
    - quiescent, 8
- évanescent, 74
- langage, 3
- mesure uniforme, 2
- mot bloquant, 11, 42
  - direction, 42
- mur, 42
  - épaisseur, 42
  - direction, 42
- nilpotent, 10
- persistant, langage, 74
- point d'équicontinuité, 11, 45
  - direction, 45
- produit cartésien, 17
- récurivement énumérable, 65
  - à droite, 66
  - à gauche, 66
- segment, 86
  - acceptable, 91
  - prédécesseur, 87
  - successeur, 87
- shift, 2
- signal, 12
- site, 6
- sous-shift, 2
  - calculable, 96
  - de type fini, 3
  - sofique, 3

# Résumé

Les automates cellulaires sont un modèle de calcul parallèle largement étudié. De nombreuses études s'intéressent à un objet classique : l'ensemble limite, qui recense les comportements asymptotiques. Dans ce mémoire, on s'intéresse à l'ensemble  $\mu$ -limite qui est une variante de l'ensemble limite traditionnel, pour laquelle on utilise une mesure sur l'espace des entrées, sélectionnant ainsi les comportements susceptibles d'apparaître arbitrairement tard et souvent. Ce nouvel ensemble a été introduit en 2000 par Petr Kurka et Alejandro Maass. On montre que sous certaines contraintes relatives à la dynamique directionnelle, l'ensemble  $\mu$ -limite peut être entièrement décrit. On classe ainsi les automates en fonction de ces contraintes. Dans le cas général, on montre l'existence d'automates cellulaires ayant comme ensembles  $\mu$ -limites un grand nombre d'ensembles complexes. On finit par montrer un théorème de Rice pour les ensembles  $\mu$ -limites d'automates cellulaires : toute propriété non triviale de ces ensembles est indécidable.

**Mots clefs** : Automates cellulaires, ensemble  $\mu$ -limite, dynamique directionnelle, équicontinuité, conséquences

# Abstract

Cellular automata are a well-known model of computation. A classical object has been often studied : the limit set which is the gathering of all asymptotic behaviors. In this thesis, we work on the  $\mu$ -limit set, which is different since we add a measure on the space of states of the system. Thus, we get a set which contains behaviors that appear arbitrarily far and often. This set was introduced in 2000 by Petr Kurka and Alejandro Maass. We study the family of all  $\mu$ -limit sets of cellular automata. We show that under some constraints concerning directional dynamics, the  $\mu$ -limit set can be entirely described. We then produce a classification of cellular automata according to these constraints. In the general case, we prove the existence of cellular automata whose  $\mu$ -limit sets are among a large set of complex sets. We finally prove Rice's theorem for  $\mu$ -limit sets : any non-trivial property is undecidable.

**Keywords** : Cellular automata,  $\mu$ -limit set, directional dynamics, equicontinuity, consequences