

Simulation Espace-Temps Réversible d'Automates Cellulaires Non-Réversibles

Jérôme Olivier DURAND-LOSE*

LABRI, Université Bordeaux I,
351, cours de la Libération,
33 405 TALENCE Cedex, FRANCE.

Présenté au GRD-PRC AMI Nice 9-11 décembre 96

Résumé

Nous rappelons brièvement les définitions des automates cellulaires (AC), de simulation et de réversibilité. En considérant les diagrammes espace-temps et non plus les itérations individuellement, nous introduisons la 'simulation espace-temps' qui se dégage de travaux récents. Nous utilisons la classe la mieux adaptée à la réversibilité: les AC partitionnés de Morita (ACP). Finalement nous construisons la simulation espace-temps d'un 1-AC quelconque par un 1-ACP réversible grâce au balayage d'un signal qui produit progressivement le diagramme espace-temps simulé. Nous expliquons comment étendre ce résultat à toute dimension.

1 Introduction

Les automates cellulaires (AC) servent de modèle pour le calcul massivement parallèle et pour des phénomènes physiques. Ils opèrent sur des matrices de dimension finie mais de taille infinie (\mathbb{Z}^d). Les éléments de cette matrice sont des *états* appartenant à un ensemble fini (\mathcal{S}). Un AC met à jour une configuration en changeant la valeur de chaque cellule selon la valeur des cellules voisines et une unique fonction locale. C'est un procédé synchrone, parallèle, local et uniforme.

Il existe plusieurs façons d'utiliser les AC pour le calcul, de 'rentrer' les données et de les récupérées [11]. Si on utilise un AC comme reconnaisseur de langages, c'est en général l'état d'une cellule bien précise qui est discriminant. Comme calculateur, c'est la valeur d'une portion de la configuration finale. Pour une simulation d'un phénomène physique, c'est aussi bien la configuration finale que les configurations intermédiaires, en tant qu'états et trace de la dynamique du système, qui sont importantes.

La réversibilité offre la possibilité de revenir en arrière sans équivoque. Elle signifie aussi la conservation de l'information et de l'énergie. Le problème de simuler de manière réversible un phénomène quelconque a toujours intéressé les chercheurs. En 1973, Bennett [2] a prouvé que toute Machines de Turing peut être simulée par une autre réversible, en temps double et avec nettoyage de la bande par un anti-calcul.

Un AC est réversible (AC-R) quand sa fonction globale \mathcal{G} est réversible et l'inverse de \mathcal{G} est elle-même la fonction globale d'un AC. La recherche sur la réversibilité des AC commença dans les années 60 quand Moore [13] et Myhill [16] démontrèrent que l'injectivité implique la surjectivité. Hedlung [7] et Richardson [17] démontrèrent indépendamment que les fonctions de $\mathcal{S}^{\mathbb{Z}^d}$ commutant avec les translations et continues pour la topologie produit sont des fonctions globales d'AC ; ceci implique en particulier qu'être injectif suffit pour être réversible. En 1977, Toffoli [18] montra que tout AC peut être simulé par un AC-R possédant une dimension de plus, prouvant ainsi l'existence d'AC-R universels pour le calcul en dimension 2 et plus. Ce n'est qu'en 1992 que leur existence en dimension 1 fut démontrée par Morita [14]. En 1972, Amoroso et Patt [1] démontrèrent que la réversibilité des 1-AC est décidable, mais ce n'est qu'en 1990 que Kari [10] montra que ce n'est plus le cas en dimension supérieure. En 1995, l'auteur [3, 4] a démontré qu'il existe des AC-R

*e-mail: jdurand@labri.u-bordeaux.fr, <http://www.labri.u-bordeaux.fr/~jdurand>

capables de simuler n'importe quel AC-R de la même dimension (supérieure ou égale à 2). Ce résultat a été étendu à la dimension 1 en 1996 [5].

Savoir s'il est possible de simuler un AC quelconque avec un AC-R de même dimension reste un problème ouvert. En 1995, Morita [15] a répondu affirmativement, tant que l'on se limitait aux configurations finies. Mais celles-ci ne sont qu'une partie des configurations récursives et nous sommes encore loin des l'ensemble des configurations infinies. De plus, la finitude est trop restrictive en physique et en mathématiques. Pour des raisons 'thermodynamiques' et de part notre expérience sur le sujet, nous ne pensons pas que des simulations de non réversibles par des réversibles de la même dimension soient possibles. Dans toutes nos simulations, nous utilisons explicitement l'AC inverse pour construire un simulateur réversible.

Généralement, simulation veut dire retrouver le même résultat quelque soient les conditions initiales. Pour les simulations entre systèmes itératifs, on sous-entend que les itérations de l'un se retrouvent dans celles de l'autre ; en particulier, que chaque itération se retrouve entièrement codée dans une seule itération, et non pas réparties sur plusieurs d'entres elles.

En calculabilité, on parle de simulation d'une machine/automate/système de réécriture/... par un autre. A partir de cela se construisent les notions d'équivalence entre les systèmes de programmation et d'universalité comme capacité de tout simuler. Au niveau des machines (dans son sens le plus général) le 'fonctionnement' se définit par induction, récurrence, itération, ... Une machine en simule une autre si elle 'passe' par les mêmes 'états' à un 'codage' près. Les termes entre parenthèse dépendent évidemment de ce que l'on manipule (Machines de Turing, λ -calcul, Réseau systolique ou de Pétri...). En complexité, on parle de réduction d'un problème à un autre. Il s'agit d'obtenir le même résultat sur les entrées valides. Là, il n'y a plus d'étapes intermédiaires, mais seulement l'entrée, la sortie, le codage et le temps. L'exécution/algorithmes est une boîte noire. Dans le premier cas, on se soucie purement du comment alors que dans la seconde c'est le pourquoi : ce que l'on veut, comment on l'exprime.

Indépendamment, de plus en plus de travaux considèrent les diagrammes espace-temps (orbite d'une configuration) dans leur ensemble et non plus les itérations séparément. Mazoyer [11] a défini un placement dynamique des signaux sur des grilles au sein de diagrammes. Heen [8, 9] a développé un positionnement statique accélérant les calculs. Les fonctions constructibles ou Fisher-constructibles [6, 12] correspondent à des propriétés géométriques de diagrammes espace-temps. L'objet construit n'est plus en sortie, mais codé au sein des itérations. Dans tous ces cas, le diagramme tout entier est le résultat et/ou la preuve. C'est sur ces observations que repose notion de simulation *espace-temps*.

Le papier s'articule comme suit : dans la sect. 2, les définitions des automates cellulaires classiques (AC) et partitionnés (ACP) sont rappelées ainsi qu'un lemme sur la réversibilité des ACP. Dans la sect. 3, nous rappelons la notion usuelle de simulation et montrons rapidement comment réduire le voisinage à $\{0, 1\}^d$ avant de définir les diagrammes espace-temps et la simulation espace-temps. Dans la sect. 4, nous construisons une simulation espace-temps d'un AC de dimension 1 (1-AC) quelconque par un 1-ACP-R. Cette construction est basée sur les allers et retours d'un signal entre deux bornes. Quand il va vers le milieu de cette portion, il est sur des cellules à des itérations de plus en plus avancées qu'il met à jour. En allant vers les extrémités, il ne met plus rien à jour et revient vers des itérations de plus en plus anciennes.

2 Définitions

Les automates cellulaires (AC) et automates cellulaires partitionnés (ACP) définissent des mises à jour parallèles et synchrones sur des matrices infinies de dimension finie d . Les intersections de la matrice sont des cellules et sont dans un *état* d'un ensemble fini \mathcal{S} . L'ensemble de toutes les configurations est $\mathcal{C} = \mathcal{S}^{\mathbb{Z}^d}$.

Automates Cellulaires

Un *automate cellulaire* de dimension d (d -AC) est défini par $(\mathcal{S}, \mathcal{N}, f)$. Son *voisinage* \mathcal{N} est une sous-ensemble fini de \mathbb{Z}^d . Sa *fonction locale* $f : \mathcal{S}^{\mathcal{N}} \rightarrow \mathcal{S}$ envoie les états d'un voisinage sur un état. La *fonction globale* $\mathcal{G} : \mathcal{C} \rightarrow \mathcal{C}$ opère sur les configurations comme suit :

$$\forall c \in \mathcal{C}, \forall x \in \mathbb{Z}^d, \mathcal{G}(c)_x = f((c_{x+\nu})_{\nu \in \mathcal{N}}) .$$

La nouvelle valeur d'une cellule ne dépend que des valeurs des cellules voisines.

Automates Cellulaires Partitionnés

Suivant la définition de Morita [14, 15], un *automate cellulaire partitionné* de dimension d (d -ACP) est défini par : $(\mathcal{S}, \mathcal{N}, \Phi)$. Les états sont des produits d'états indexés par le voisinage : $\mathcal{S} = \prod_{\nu \in \mathcal{N}} \mathcal{S}^{(\nu)}$. La composante selon ν d'un état s est notée $s^{(\nu)}$. La fonction Φ opère sur \mathcal{S} et la fonction locale f se définit par :

$$\forall c \in \mathcal{C}, \forall x \in \mathbb{Z}^d, f(c)_x = \Phi \left(\prod_{\nu \in \mathcal{N}} c_{x+\nu}^{(\nu)} \right) .$$

Autrement dit, chaque état est le produit de ce qui doit être envoyé autour. La fonction locale opère sur ce qui reste et ce qui est reçu. La cellule n'a qu'une connaissance partielle de son propre état comme des états de ses voisines, comme illustré sur la fig. 1.

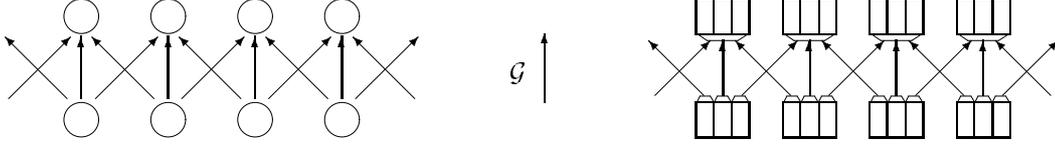


Figure 1: Mise à jour des AC et des ACP.

Réversibilité

Un AC (ACP) est *réversible* si sa fonction globale \mathcal{G} est bijective et son inverse \mathcal{G}^{-1} est la fonction globale d'un AC (ACP). Nous notons AC-R (ACP-R) la classe des AC (ACP) réversibles. Pour les ACP le lemme suivant est vrai en toute dimension :

Lemme 1 (Morita) *Un ACP est réversible si et seulement si sa fonction locale Φ est une permutation, ce qui est décidable.*

Preuve Si Φ est une permutation, alors l'ACP inverse est $(\prod_{\nu \in -\mathcal{N}} \mathcal{S}^{(-\nu)}, -\mathcal{N}, \Phi^{-1})$ où $-\mathcal{N} = \{-\nu | \nu \in \mathcal{N}\}$. Φ est défaite et les morceaux renvoyés d'où ils viennent.

Sinon, Φ opérant sur un ensemble fini, elle n'est pas injective, il est alors facile de construire 2 configurations ayant la même image.

La décidabilité vient de la finitude de \mathcal{S} . □

En 1990, Kari [10] a prouvé que la réversibilité des AC de dimension 2 et plus n'était pas décidable. En ce qui concerne la réversibilité, AC et ACP sont donc fondamentalement différents.

3 Simulation

3.1 Approche Itérative

Les AC étant des fonctions de \mathcal{C} , on désire retrouver une configuration entièrement codée dans une autre configuration. La définition est celle de Toffoli [18].

Définition 2 Un automate **A** *simule (itérativement)* un autre **B** s'il existe 3 fonctions $\psi : \mathcal{C}_B \times \mathbb{N} \rightarrow \mathbb{N}$, $\alpha : \mathcal{C}_B \rightarrow \mathcal{C}_A$ et $\beta : \mathcal{C}_A \rightarrow \mathcal{C}_B$ telles que :

$$\forall c \in \mathcal{C}_B, \forall t \in \mathbb{N}, \mathcal{G}_B^t(c) = \beta \circ \mathcal{G}_A^{\psi(c,t)} \circ \alpha (c) .$$

Les fonction ψ , α et β doivent être de complexité inférieure aux AC pour assurer qu'elles ne font pas le calcul. Dans la pratique, ce sont de projections, injections canoniques... et le problème de la complexité ne se pose pas. On accepte que, à c fixé, $\psi(c,t)$ ne soit pas définie pour tout t tant qu'elle le reste pour une infinité de t . Ceci est nécessaire pour les accélérateurs : par exemple, pour simuler $3n$ itérations en n itérations, il y a $2n$ itérations qui ne peuvent être définies.

La simulation est en *temps linéaire* τ si, pour toute configuration c , $\psi(c, t) = \tau t$. Si $\tau=1$, la simulation est en *temps réel*. Comme les d -ACP sont des d -AC, ils peuvent être simulés par les d -AC. De même, les d -ACP-R peuvent être simulés par des d -AC-R. Dans [5], il y a une construction de simulation de d -AC (d -AC-R) par d -ACP (d -ACP-R) en temps linéaire.

Le lemme suivant sert aussi d'exemple de simulation.

Lemme 3 *Tout d -AC peut être simulé par un d -AC de voisinage $\{-1, 0, 1\}^d$ en temps réel.*

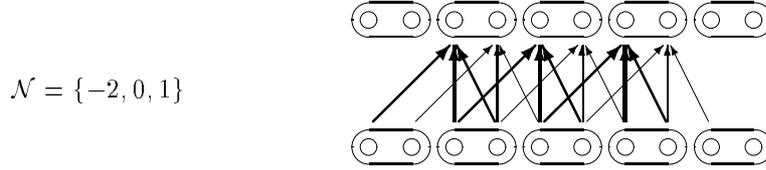


Figure 2: Regroupement par 2 cellules.

Preuve Il suffit de regrouper plusieurs cellules en une seule. Soit $A = (\mathcal{S}_A, \mathcal{N}, f_A)$ le plus général des d -AC. Soit r le rayon de A , c'est-à-dire le maximum des valeurs absolues des composantes des éléments du voisinage \mathcal{N} . Nous définissons $B = (\mathcal{S}_A^d, \{-1, 0, 1\}^d, f_B)$. La fonction α est alors définie par $(\alpha(c))_i = c_{rx+i}$, $\beta = \alpha^{-1}$ et f_B par induction à partir de f_A comme sur la fig. 2. \square

3.2 Approche Espace-Temps

Définition 4 Un *diagramme espace-temps* \mathbb{A} est l'orbite d'une configuration par un AC.

En d'autres termes, soit \mathcal{G} la fonction globale d'un d -AC A dont l'ensemble d'états est \mathcal{S} et une configuration initiale $c \in \mathcal{S}^{\mathbb{Z}^d}$. Le diagramme espace-temps associé est $\mathbb{A} : \mathbb{Z}^d \times \mathbb{N} \rightarrow \mathcal{S}$ tel que : $\mathbb{A}_{(x,t)} = (\mathcal{G}^t(c))_x$. Il se note $\mathbb{A} = (\mathcal{G}, c)$ ou (A, c) .

Un diagramme espace-temps \mathbb{A} est 'inséré' dans un autre diagramme espace-temps \mathbb{B} quand on peut 'reconstruire' \mathbb{A} à partir de \mathbb{B} et de la façon dont il est 'disposé' dans \mathbb{B} . Mathématiquement, ceci est relativement compliqué à exprimer, nous conseillons de regarder la simulation de la section suivante et [8] qui en sont des exemples.

Définition 5 Un diagramme espace-temps $\mathbb{A} = (A, c)$ est *inséré* dans un autre $\mathbb{B} = (B, d)$ quand il existe trois fonctions $\chi : \mathbb{Z}^d \times \mathbb{N} \rightarrow \mathbb{N}$, $\alpha : \mathcal{C}_B \rightarrow \mathcal{C}_A$ et $\beta : \mathcal{C}_A \rightarrow \mathcal{C}_B$ telles que :

- $d = \alpha(c)$;
- $\forall (x, t) \in \mathbb{Z}^d \times \mathbb{N}$, soit e la configuration de A telle que $e_x^t = \mathbb{A}_{x, \chi(x,t)}$;
- $\forall t \in \mathbb{N}$, $\mathcal{G}_B^t(c) = \beta(e^t)$.

La configuration c est codée dans d grâce à α . Pour retrouver une image itérée de c , χ dit quelle itération considérer pour chaque cellule, et β décode sur la configuration ainsi créée. Comme précédemment, il se peut que e^t ne soit pas définie pour tout t , et les fonctions χ , α et β doivent être de complexité inférieure aux diagrammes.

Définition 6 Un AC A *simule en espace-temps* un AC B quand tout diagramme espace-temps de B peut être inséré dans un diagramme espace-temps de A , avec les mêmes fonctions α et β .

A priori, il faut une fonction χ pour chaque configuration à l'origine d'un diagramme. Si celle-ci ne dépend que du temps et de la configuration initiale c , on retrouve la simulation itérative. Cette notion englobe la précédente et est aussi transitive.

4 Simulation Espace-temps par AC Réversible

Dans cette section, nous démontrons le lemme suivant en construisant la simulation.

Lemme 7 *Tout d -AC de voisinage $\{-1, 0, 1\}^d$ peut être simulé en espace-temps par un d -ACP-R.*

Preuve La preuve n'est détaillée qu'en dimension 1. Nous expliquons plus loin comment l'appliquer à n'importe quelle dimension. Soit $A=(\mathcal{S}_A, \{-1, 0, 1\}, f)$ le plus général des 1-AC avec ce voisinage. Nous construisons $P=(\mathcal{S}_P, \{-1, 0, 1\}, \Phi)$, un 1-ACP-R qui le simule en espace-temps.

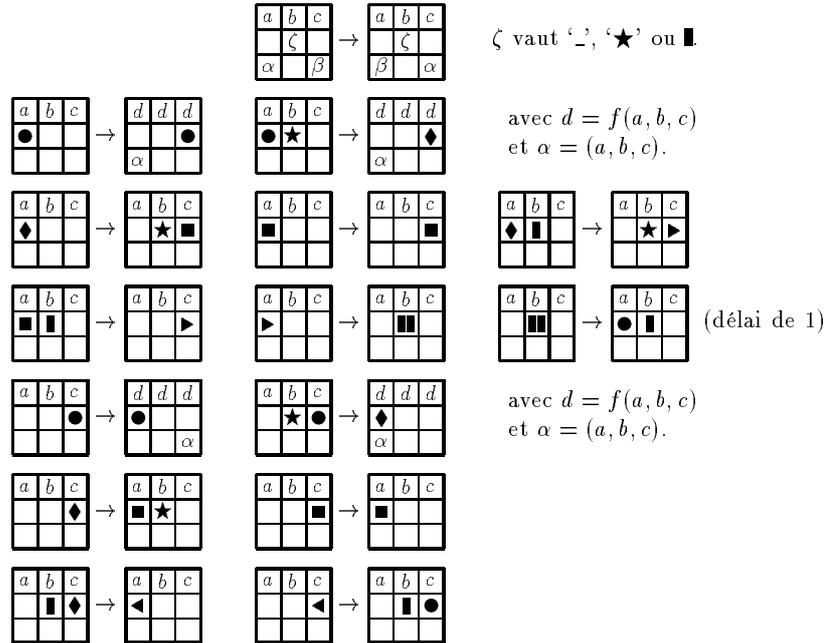
La construction est organisée en 3 couches. L'ensemble des états de P est indiqué sur la fig. 3. Pour un AC à m états, il faut $100m^3(m^3+1)^2$ états ce qui représente un fort accroissement de la taille de la table et de la complexité.

-1	0	1
\mathcal{S}_A	\mathcal{S}_A	\mathcal{S}_A
◀ ■ ● ◆ -	▬ ▬ ▬ ★ -	▶ ■ ● ◆ -
$\mathcal{S}_A^3 \cup \{-\}$	-	$\mathcal{S}_A^3 \cup \{-\}$

Figure 3: Ensemble des états de P : \mathcal{S}_P .

Les cellules sont représentées par des matrices 3×3 comme sur la fig. 4. La couche du haut contient des états de P : elle contient une configuration de A . Celle du milieu contient des signaux qui font la mise à jour. Celle du bas sert à conserver l'information nécessaire à la réversibilité.

Tout est dirigé par la présence de signaux sur la couche centrale : 2 ▬ délimitent la partie de la configuration qui n'est plus à l'itération 0 (i.e., configuration initiale). Un signal fait des allers et retours entre les deux ▬, il vaut tantôt ●, tantôt ■. Sur son passage, le signal ● met à jour les cellules alors que ■ ne fait que passer. La partie à mettre à jour se termine par ★.



a, b, c et d appartiennent à \mathcal{S} .
 α et β appartiennent à $\mathcal{S}_A^3 \cup \{-\}$.

Figure 4: Ensemble des règles de Φ .

Les règles de transitions sont décrites par la fig. 4. La règle sur la première ligne correspond à l'absence de signal se déplaçant, les deux sous-états de la dernière ligne sont échangés ce qui produit un effet de tapis

roulant. En effet, dès que l'on met quelque chose sur la dernière ligne, elle se déplace d'une case par unité de temps. Ceci sert à évacuer les données nécessaires à la réversibilité mais encombrantes pour la suite du calcul. Les règles correspondant à la mise à jour sont sur les lignes 2 et 5. Elles sont injectives, Φ peut donc être complétée bijectivement ; \mathbf{B} est alors réversible (lem. 1).

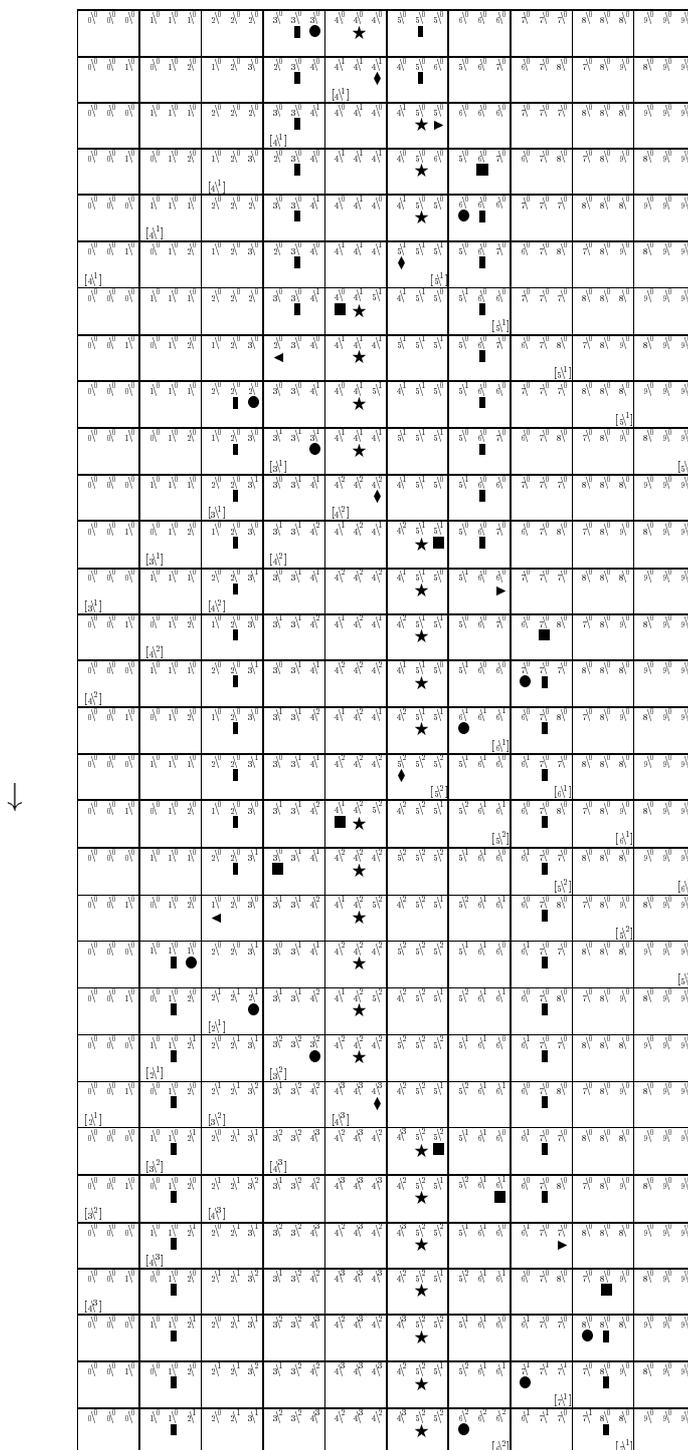


Figure 5: 26 premières itérations de la simulation espace-temps.

Les 26 premières itérations sont détaillées sur la fig. 5. Pour le diagramme inséré, la valeur de la cellule x au temps t est noté x^t et l'information qu'il a fallu pour l'obtenir $[x^t]$ (i.e., $[x^t] = (x-1)^{t-1}, x^{t-1}, x+1)^{t-1}$). Les cellules contiennent tantôt trois fois la même valeur soit trois valeurs voisines à la même itération, sinon elle contiennent un mélange sur plusieurs itérations. Une cellule ne peut être mise à jour que quand $[x^t]$ est dans la cellule. Par induction à partir de la fig. 5, il apparaît que la possibilité de mettre à jour une cellule dépende uniquement de la parité de la somme des ordres des itérations du simulé et du simulant. Le signal ● fait un va et vient et il y a un retard à droite pour conserver cette parité.

La configuration initiale est sur la première ligne de la fig. 5. La valeur de chaque cellule est copiée en trois exemplaires dans la couche du haut. Il y a ■, ★ et ■ dans les centre de trois cellules voisines et ● avec le ■ de gauche.

Les ■ s'écartent progressivement de ★ qui oscille au milieu. Depuis un ■, le signal est un cercle ●, il met à jour les cellules jusqu'à atteindre ★; après c'est un carré ■ qui se contente de rejoindre l'autre ■.

A chaque mise à jour, 3 valeurs ($[x^t]$) sont 'consommées' et copiées en bas pour la réversibilité. De même, la nouvelle valeur est créée en 3 exemplaires car elle servira pour la cellule et ses 2 voisines.

La pérennité du va et viens de ● mettant à jour au bon moment se déduit du fait que tout se passe localement, avec un rayon 1. Toutes les étapes nécessaire à l'induction se trouve dans les 2 allers et retours et demi de ● sur la fig. 5.

La fig. 6 décrit l'évacuation des $[x^t]$ ainsi que le mouvement des signaux pour les 170 premières itérations. Les valeurs évacuées n'interagissent jamais. Le signal fait n mises à jour dans un sens puis n au retour, puis $n + 1$ et $n + 1$...

La fig. 7 montre comment est engendré le diagramme espace-temps simulé. Les numéros correspondent à l'ordre de passage de ●; les flèches correspondent au sens de parcours. L'aspect construction en oignon (couche après couche) est particulièrement visible.

Cette construction se généralise à n'importe quelle dimension plus grande que 1. Le recourbement du diagramme espace temps se fait toujours selon la première direction. Suivant cette direction, la dynamique est exactement celle décrite au-dessus. Par contre, selon les autres directions, on retrouve identiquement les mêmes signaux et numéro d'itérations. La mise à jour est toujours conditionnée par la parité de la somme des ordres des itérations du simulé et du simulant. Il y a une infinité de ■, ● et de ★, ils sont répartis selon des hyperplans perpendiculaires à la première direction et exactement synchronisés. □

Théorème 8 *Tout d -AC peut être simulé en espace-temps par un d -AC-R.*

Preuve Cela découle du fait que tout AC peut être simulé en temps réel par un d -AC de voisinage $\{-1, 0, 1\}$ (lem. 3), du lemme 7 précédent et du fait que les d -ACP-R sont des d -AC-R. □

5 Conclusion

Nous avons donc réussi à simuler en espace-temps un AC quelconque par un AC réversible. Il existe des d -AC-R capables se simuler (itérativement) tous les d -AC-R sur toutes les configurations [5].

Théorème 9 *Il existe des d -AC-R capables de simuler en espace temps tout d -AC.*

Malheureusement, le diagramme espace-temps simulé est tordu en \vee ce qui rend difficilement accessible des propriétés géométrique comme la présence dans un état donné de la cellule 0 à un temps donné (Fisher-constructibilité). D'un autre coté, cela permet de dire que si une fonction est Fisher-constructible, alors la même retardé de certains polynômes du second degré en l'ordre de l'itération, l'est aussi.

Dans notre simulation espace-temps, il n'est pas possible de revenir en arrière car il faut que toutes les configurations précédentes soient dans les configurations. Ceci demandera un α trop compliqué (sa complexité est limitée), il y a de plus des problèmes liés à l'existence et à la multiplicité des configurations antécédentes.

Dans notre construction, il faut un temps infini pour engendré entièrement une itération. Quand la partie significative de la configuration occupe une partie finie de l'espace, la partie significative de chaque itération est obtenue en temps fini. Ceci n'est pas plaisant pour une configuration quelconque. La notion de

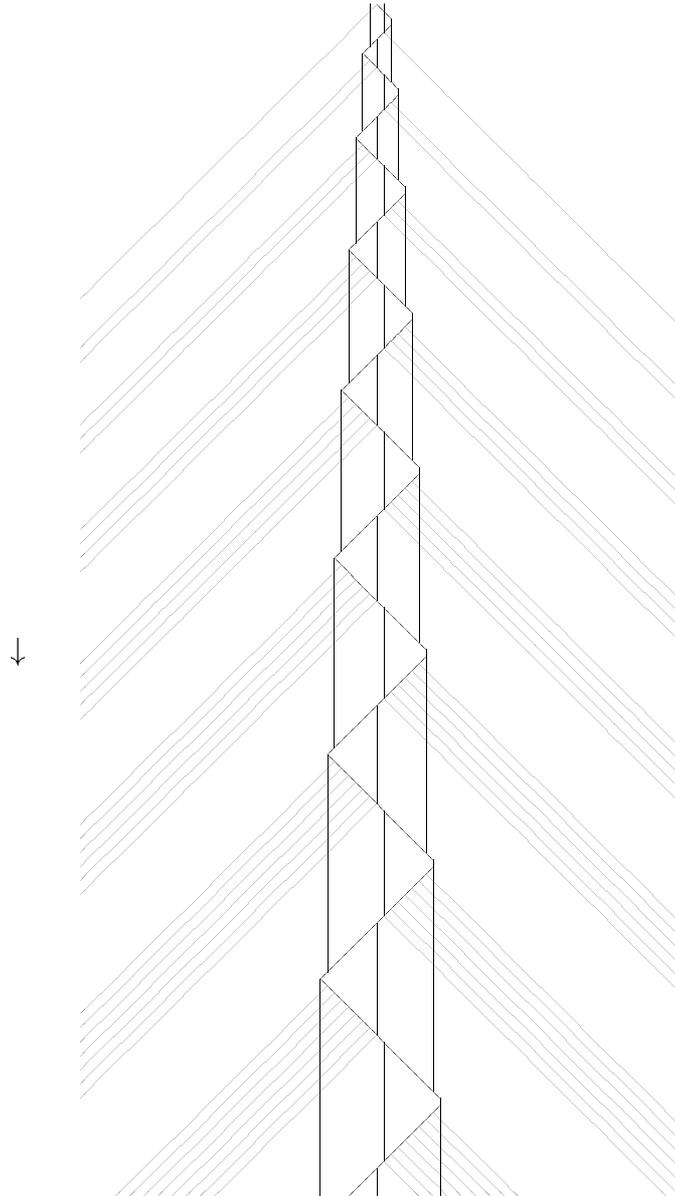


Figure 6: Schéma des itérations.

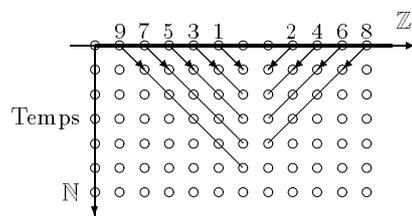


Figure 7: Engendrement du diagramme.

simulation en espace-temps doit avoir moins de propriétés que la simulation usuelle car plus large, peut-être y-a-t-il des différences comme entre limites simple et uniforme en mathématiques.

Il serait intéressant de savoir dans quelle mesure les techniques et résultats développés ici peuvent être adaptés au cas où l'on impose à la fonction χ d'être à différence bornée à t fixé. Par exemple, s'il existe un entier c tel que $\forall x, y, \forall t, |\chi(x, t) - \chi(y, t)| \leq c$ cela permet-il de dire qu'il y a une simulation au sens premier au moyen de regroupements?

Références

- [1] S. Amoroso and Y. Patt. Decision procedure for surjectivity and injectivity of parallel maps for tessellation structure. *Journal of Computer and System Sciences*, 6:448–464, 1972.
- [2] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 6:525–532, 1973.
- [3] J. O. Durand-Lose. Reversible cellular automaton able to simulate any other reversible one using partitioning automata. In *LATIN '95*, number 911 in Lecture Notes in Computer Science, pages 230–244. Springer-Verlag, 1995.
- [4] J. O. Durand-Lose. *Automates Cellulaires, Automates à Partitions et Tas de Sable*. PhD thesis, LaBRI, 1996. In French.
- [5] J. O. Durand-Lose. Intrinsic universality of a 1-dimension reversible cellular automaton. Technical Report 1136-96, LaBRI, URA CNRS 1304, Université Bordeaux I, 351, cours de la Libération, 33 405 TALENCE Cedex, FRANCE, 1996. Accepted to STACS '97.
- [6] P. C. Fisher. Generation of primes by a one-dimension real-time iterative array. *Journal of the ACM*, 12(3):388–394, 1965.
- [7] G. A. Hedlung. Endomorphism and automorphism of the shift dynamical system. *Mathematical System Theory*, 3:320–375, 1969.
- [8] O. Heen. A linear speed-up theorem for cellular automata synchronizers and applications. To appear in *Theoretical Computer Science*.
- [9] O. Heen. *Economie de Ressources sur Automates Cellulaires*. PhD thesis, LITP, IBP, Université Paris 7, 1996. In French.
- [10] J. Kari. Reversibility of 2D cellular automata is undecidable. *Physica D*, 45:379–385, 1990.
- [11] J. Mazoyer. Computations on one dimensional cellular automata. Technical Report 94-52, LIP, ENS Lyon, 46 allée d'Italie, 69 364 LYON7, 1994. To appear in *Mathematic and Computer Science some interactions*, Balsep publication.
- [12] J. Mazoyer. Signals in one dimensional cellular automata. Technical Report 94-50, LIP, ENS Lyon, 46 allée d'Italie, 69 364 LYON7, 1994.
- [13] E. Moore. Machine models of self-reproduction. In *Proceeding of Symposium on Applied Mathematics*, volume 14, pages 17–33, 1962.
- [14] K. Morita. Computation-universality of one-dimensional one-way reversible cellular automata. *Information Processing Letters*, 42:325–329, 1992.
- [15] K. Morita. Reversible simulation of one-dimensional irreversible cellular automata. *Theoretical Computer Science*, 148:157–163, 1995.
- [16] J. Myhill. The converse of Moore's garden-of-eden theorem. In *Proceedings of the Symposium of Applied Mathematics*, number 14, pages 685–686, 1963.
- [17] D. Richardson. Tessellations with local transformations. *Journal of Computer and System Sciences*, 6:373–388, 1972.
- [18] T. Toffoli. Computation and construction universality of reversible cellular automata. *Journal of Computer and System Sciences*, 15:213–231, 1977.