

1. Introduction

Votre treillis à 4 couleurs est un système de réécriture locale. Chaque cellule du pavage est déterminée par une règle du type $(x_1, y_1)(x_2, y_2) \rightarrow (x_1, y_2)$. C'est un automate cellulaire élémentaire, mais il s'applique à une grille triangulaire dont les deux axes (x et y) portent une information binaire (primalité).

Pour répondre à votre question sur les **chaînes de Markov**, il faut distinguer deux choses :

1. modéliser le processus de génération du pavage (la règle de pavage) par une chaîne de Markov.
2. modéliser l'entrée elle-même (la suite des primalités) par une chaîne de Markov.

2. La règle de pavage comme chaîne de Markov (déterministe)

Une chaîne de Markov est un processus stochastique à mémoire finie : l'état futur ne dépend que de l'état présent. Si on fixe une colonne du pavage (c'est-à-dire une séquence finie de couleurs le long de l'axe y), alors la règle

$$(x_1, y_1)(x_2, y_2) \rightarrow (x_1, y_2)$$

permet de calculer une colonne de manière déterministe à partir de la colonne qui la précède. C'est un shift de type fini (ou un sous-shift).

On peut donc voir l'évolution du pavage colonne par colonne comme une chaîne de Markov d'ordre 1 (ou d'ordre supérieur si on tient compte de la mémoire) dont les états sont les colonnes possibles (des mots binaires de longueur fixée, ou infinis). La matrice de transition est alors déterministe (probabilités 0 ou 1) : de la colonne C_i , on passe à la colonne C_{i+1} .

Conclusion : votre treillis est un sous-shift sofique (un ensemble de colonnes défini par des contraintes locales). C'est un objet bien étudié en dynamique symbolique, et il peut être représenté par un graphe de transitions (un automate) dont les états sont les colonnes.

3. L'entrée (la suite de primalité) comme chaîne de Markov

C'est là où le bât blesse. Vous avez raison de dire que votre treillis a besoin, en entrée, de la suite des primalités sur l'axe des x (et sur la diagonale (ou sur l'axe des y)). La question est : peut-on remplacer cette suite déterministe par un processus aléatoire markovien ?

- si vous connaissez la suite de primalité (vous l'avez calculée), vous pouvez l'injecter dans le pavage et obtenir un pavage déterministe. C'est un pavage codé par une suite donnée.

- si vous voulez générer une suite de primalité par une chaîne de Markov, vous êtes face à un problème de modélisation. Une chaîne de Markov à mémoire finie (par exemple, une chaîne de Bernoulli, ou une chaîne d'ordre 1) produit des suites dont la complexité de sous-mots est linéaire (ou exponentielle avec une base constante). Or la suite des nombres premiers a une **complexité de sous-mots** qui croît plus vite que toute fonction linéaire (c'est une conséquence du théorème des nombres premiers).

En d'autres termes :

La suite des primalités ne peut pas être produite par une chaîne de Markov à mémoire finie.

C'est exactement la même limitation que pour les automates finis : une chaîne de Markov à états finis est un cas particulier de machine à états finis (avec des probabilités de transition). Elle ne peut pas générer une suite aussi complexe que la suite des nombres premiers.

4. Et si on autorise une mémoire infinie ?

On pourrait imaginer une chaîne de Markov d'ordre variable (par exemple, une chaîne de Markov d'ordre non stationnaire ou un processus de renouvellement) dont la mémoire croît avec le temps. Mais alors on sort du cadre classique des chaînes de Markov (mémoire bornée). C'est un autre modèle de calcul, plus puissant, mais qui n'utilise plus un automate fini.

On pourrait aussi utiliser une chaîne de Markov cachée, où l'état caché encode un compteur qui permet de détecter les nombres premiers. Mais ce compteur nécessite une mémoire qui croît avec la taille des nombres, ce qui n'est pas un processus fini.

5. Pistes pour une modélisation probabiliste de votre treillis

Si vous voulez tout de même utiliser un modèle probabiliste, voici deux approches possibles :

1. **modèle de percolation** : au lieu de prendre la suite exacte des primalités, on pourrait la remplacer par une suite de bits indépendants et identiquement distribués avec une probabilité $p = \frac{1}{\log n}$ (qui est la densité locale des nombres premiers). On obtiendrait alors un pavage aléatoire dont on pourrait étudier les propriétés statistiques (par exemple, la probabilité qu'une cellule soit de type $(0, 0)$ se comporte comme p^2). C'est une modélisation asymptotique, pas exacte.
2. **processus de renouvellement** : on sait que les écarts entre nombres premiers suivent des lois statistiques (conjecture de Cramér). On pourrait simuler une suite de primalités en tirant aléatoirement les écarts entre nombres premiers successifs selon une loi appropriée (par exemple, exponentielle de paramètre $\log n$). Cela donnerait une chaîne de Markov d'ordre 1 sur les écarts, avec une mémoire bornée. Mais ce ne serait pas la vraie suite des nombres premiers ; ce serait une approximation probabiliste, utile pour des simulations numériques mais pas pour une preuve mathématique.

6. Conclusion

- oui, votre treillis peut être vu comme un sous-shift (un système dynamique symbolique) dont les transitions sont déterministes. C'est un modèle de calcul local, proche d'un automate cellulaire.
- non, on ne peut pas modéliser exactement l'entrée (la suite de primalité) par une chaîne de Markov à mémoire finie, pour la même raison qu'elle n'est pas automatique.
- en revanche, si on accepte une modélisation probabiliste approchée, on peut utiliser des chaînes de Markov (ou des processus de renouvellement) pour générer des suites de bits ayant des propriétés statistiques similaires à celles des nombres premiers, et étudier le comportement moyen du pavage.

C'est un domaine de recherche actif : on appelle cela la théorie des nombres probabiliste ou la modélisation stochastique des nombres premiers. Les chaînes de Markov sont un outil parmi d'autres (processus de Poisson, processus de renouvellement, etc.) pour simuler des suites de primalité, mais elles ne remplacent pas la suite exacte.