

1. Introduction et règles de réécriture

Le document intitulé “*Espace des nombres premiers et pavage apériodique du plan par des triminos bicolores*” <https://denisevellachemla.eu/pavages3.pdf> (mai 2017), propose une approche topologique fascinante. Elle convertit un dictionnaire de 16 règles de réécriture (ex. $aa \rightarrow a$, $cb \rightarrow d$) en un problème de pavage d’un plan euclidien.

2. Triminos de Wang et apériodicité

L’utilisation de triminos bicolores contraint les orientations et force des alignements “en diagonale”. Cette restriction géométrique mime le comportement des contraintes d’exclusion mutuelle du crible arithmétique.

Trouver une configuration de pavage globale valide et stable équivaut à identifier des chemins admissibles dans l’espace des phases, offrant une interprétation géométrique non commutative (en référence aux travaux d’Alain Connes) de la distribution des décomposants.

3. Exploration locale

Le programme Python ci-dessous simule le dictionnaire des règles de réécriture pour vérifier les transitions de caractères aux frontières des tuiles.

```
# -*- coding: utf-8 -*-
def simuler_regles_pavage():
    print("—— Piste 10 : Validation des regles de reecriture de triminos ——")
    regles = {
        ('a', 'a'): 'a', ('a', 'b'): 'b', ('a', 'c'): 'a', ('a', 'd'): 'b',
        ('b', 'a'): 'a', ('b', 'b'): 'b', ('b', 'c'): 'a', ('b', 'd'): 'b',
        ('c', 'a'): 'c', ('c', 'b'): 'd', ('c', 'c'): 'c', ('c', 'd'): 'd',
        ('d', 'a'): 'c', ('d', 'b'): 'd', ('d', 'c'): 'c', ('d', 'd'): 'd'
    }
    test_couples = [('a', 'b'), ('c', 'b'), ('d', 'd')]
    for c in test_couples:
        res = regles.get(c, regles.get((c[0], c[1][0]), 'Inconnu')) if len(c[1])>1 else 'Inconnu'
        print(f"Tuile gauche '{c[0]}' + Tuile droite '{c[1]}' -> Transition frontiere : '{res}'")

if __name__ == "__main__":
    simuler_regles_pavage()
```

Son résultat d'exécution

```
— Piste 10 : Validation des regles de reecriture de triminos —  
Tuile gauche 'a' + Tuile droite 'b' -> Transition frontiere : 'b'  
Tuile gauche 'c' + Tuile droite 'b' -> Transition frontiere : 'd'  
Tuile gauche 'd' + Tuile droite 'd' -> Transition frontiere : 'Inconnu'
```