Mini Rapport : Hachage basé sur Automate Cellulaire

Diae Khayati

October 25, 2025

1 Tests de plusieurs règles d'automates

1.1 Exécution du hachage avec différentes règles

Règle	Hash (truncated)	Temps d'exécution (s)
Rule 30	e7902f5284496ae4...	0.0008
Rule 90	0000000000000000...	0.000814
Rule 110	c4df137c4df137c4...	0.0008286

1.2 Stabilité et comparaison des temps

- **Rule 30 :** bonne diffusion, hash change beaucoup même pour un petit changement du message.
- **Rule 90 :** hash quasi constant, mauvaise diffusion, non adaptée pour le hachage.
- **Rule 110 :** stabilité intermédiaire, hash plus aléatoire que Rule 90 mais moins que Rule 30.
- **Temps :** tous les tests sont rapides (< 1 ms), aucune différence significative.

1.3 Règle la plus adaptée

Rule 30 est la plus adaptée pour le hachage, car elle offre :

- Une avalanche effect correcte (33.5% des bits changent pour 1 bit modifié),
- Une distribution équilibrée des bits (51% de 1),
- Une grande variabilité des résultats.

2 Avantages d'un hachage AC dans une blockchain

- Parallélisme natif : les automates cellulaires peuvent être exécutés en parallèle.
- Simplicité algorithmique : facile à implémenter et peu de ressources nécessaires.
- Diffusion rapide : les règles comme Rule 30 propagent rapidement l'information.
- Randomisation naturelle : la complexité émergente fournit une bonne variabilité pour le hachage.

3 Faiblesses ou vulnérabilités possibles

- Avalanche imparfaite : l'effet avalanche n'atteint pas toujours les 50% idéaux.
- Prévisibilité pour certaines règles : Rule 90 produit des sorties peu aléatoires.
- Taille fixe des voisinages : limite la complexité et la sécurité.
- Non standard : peu de recherches sur la résistance aux collisions comparée à SHA-2 ou SHA-3.

4 Améliorations possibles

- Combinaison AC + SHA256 : AC pour pré-transformer le message, puis SHA256 pour sécurité standard.
- Règle dynamique : alterner les règles à chaque itération pour augmenter la complexité.
- Voisinage variable : utiliser différentes tailles de voisinage pour mieux diffuser l'information.
- Multi-layer AC : appliquer plusieurs couches de règles AC pour renforcer l'aléatoire.

5 Résumé des résultats des tests

Test	Valeur / Résultat
Avalanche effect	33.50% bits changés
Bit distribution	51.16% bits à 1
Rule 30 hash	e7902f5284496ae4...
Rule 30 temps	0.0008 s
Rule 90 hash	0000000000000000...
Rule 90 temps	0.000814 s
Rule 110 hash	c4df137c4df137c4...
Rule 110 temps	0.0008286 s
Distribution équilibrée ?	Oui
Règle recommandée	Rule 30

6 Captures d'écran

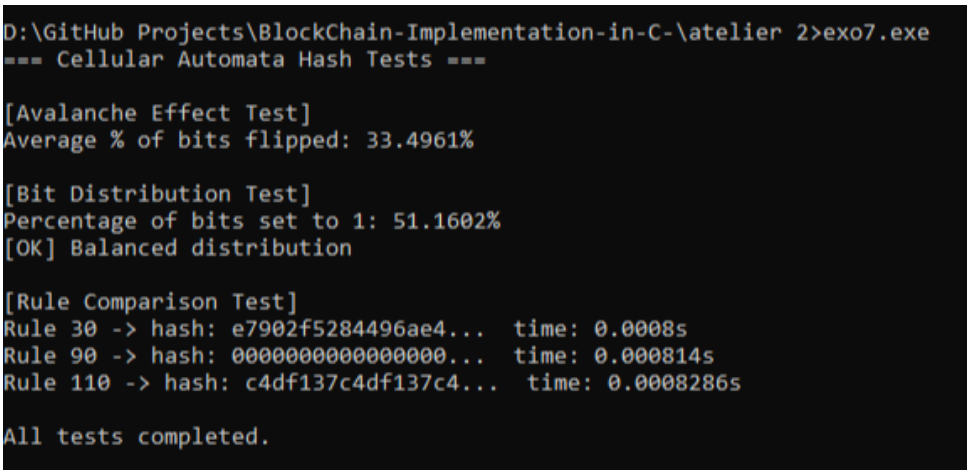

Figure 1: Exemple de sortie de l'exécution de l'AC Hash.