

1. Résumé

Nous discutons ici de la différence fondamentale entre une définition récursive (calculable pour chaque n) et la reconnaissance par un automate fini (mémoire bornée).

La récurrence de Giard pour la fonction $\sigma(n)$ est un exemple de calculabilité, mais elle n'implique pas que la suite de primalité soit reconnaissable par un automate fini.

Le théorème de Cobham (1969) fournit un critère puissant pour montrer qu'une suite n'est pas automatique.

2. Introduction

Dans nos échanges précédents, vous avez soulevé une question cruciale : la récurrence de Giard, qui définit la somme des diviseurs $\sigma(n)$ de manière récursive, ne contredit-elle pas l'affirmation selon laquelle la primalité n'est pas reconnaissable par un automate fini ?

Cette question touche à une distinction fondamentale en théorie de la calculabilité et en théorie des automates. Nous allons ici clarifier cette distinction et introduire le théorème de Cobham, qui donne un critère définitif pour savoir si une suite peut être reconnue par un automate fini.

3. La récurrence de Giard : un exemple de calculabilité

La formule de Giard, présente dans l'OEIS pour la suite A000203 (la somme des diviseurs), s'écrit :

$$\sigma(1) = 1, \\ \sigma(n) = \frac{12}{n^2(n-1)} \sum_{k=1}^{n-1} (5k(n-k) - n^2) \sigma(k) \sigma(n-k).$$

Cette formule permet de calculer $\sigma(n)$ pour chaque n fixé, en utilisant les valeurs précédentes $\sigma(1), \dots, \sigma(n-1)$.

C'est une définition récursive qui donne un algorithme effectif pour calculer $\sigma(n)$ en un nombre fini d'étapes.

3.1. Caractérisation des nombres premiers

On sait que :

$$n \text{ est premier} \iff \sigma(n) = n + 1.$$

Ainsi, la primalité de n est décidable à partir de $\sigma(n)$.

4. Pourquoi la récurrence de Giard n'implique pas un automate fini

1

Calcul récursif (formule de Giard)	Reconnaissance par automate fini
Pour un n fixé, on dispose d'un algorithme qui termine en un nombre fini d'étapes et donne $\sigma(n)$.	On considère une suite infinie de bits (par exemple, la suite des primalités). Un automate fini doit lire cette suite bit par bit, de gauche à droite, et décider en temps réel si elle vérifie une certaine propriété, avec une mémoire bornée (un nombre fini d'états).
La récurrence utilise les valeurs de $\sigma(k)$ pour tous les $k < n$. Pour n grand, cela demande de stocker et de manipuler une quantité d'information qui croît avec n .	L'automate ne peut stocker qu'une quantité constante d'information, indépendante de la longueur de la suite lue.
C'est un algorithme au sens de Church-Turing, exécutable sur une machine de Turing avec une mémoire illimitée.	C'est un modèle de calcul très restreint, équivalent aux expressions régulières.

La formule de Giard est un algorithme qui, pour chaque n , calcule $\sigma(n)$ en utilisant toutes les valeurs précédentes. Mais pour décider si tous les entiers sont premiers, un automate fini devrait pouvoir traiter la suite infinie des primalités avec une mémoire bornée, ce qui est impossible car la mémoire nécessaire pour appliquer la récurrence croît avec n .

5. Le théorème de Cobham (1969)

Le théorème de Cobham est un résultat fondamental de la théorie des suites automatiques. Il donne une condition nécessaire pour qu'une suite soit reconnaissable par un automate fini dans deux bases différentes.

5.1. Définitions préliminaires

Définition 1. Deux entiers $k \geq 2$ et $\ell \geq 2$ sont dits multiplicativement indépendants s'il n'existe pas d'entiers positifs a, b tels que $k^a = \ell^b$. Autrement dit, $\frac{\log k}{\log \ell}$ est irrationnel [1, 2].

Définition 2. Une suite $(s(n))_{n \geq 0}$ est dite k -automatique s'il existe un automate fini déterministe avec sortie qui, pour chaque n , lit l'écriture en base k de n et produit $s(n)$ [2].

1. DVC : ce titre de section est mal formulé : n'implique pas de pouvoir faire traiter la suite booléenne des caractères de primalité par un automate fini.

5.2. Énoncé du théorème

Théorème 3 (Cobham, 1969). *Soient k et ℓ deux entiers multiplicativement indépendants. Si une suite $s = (s(n))_{n \geq 0}$ est à la fois k -automatique et ℓ -automatique, alors s est ultimement périodique [1].*

En d’autres termes, une suite non ultimement périodique ne peut pas être reconnue par des automates finis dans deux bases multiplicativement indépendantes [2].

5.3. Références

[1] A. Cobham, *On the base-dependence of sets of numbers recognizable by finite automata*, Math. Systems Theory **3** (1969), 186–192.

[2] J.-P. Allouche et J. Shallit, *Automatic Sequences : Theory, Applications, Generalizations*, Cambridge University Press, 2003.

6. Application à la suite de primalité

La suite caractéristique des nombres premiers, notée $(p_n)_{n \geq 0}$ avec $p_n = 1$ si n est premier et 0 sinon, **n’est pas** ultimement périodique (elle contient des nombres premiers arbitrairement grands, et des intervalles de composés arbitrairement longs).

D’après le théorème de Cobham, si elle était k -automatique pour un certain k , elle ne pourrait pas être ℓ -automatique pour un ℓ multiplicativement indépendant. Mais surtout, il est connu qu’elle n’est **automatique dans aucune base** : c’est une conséquence du théorème des nombres premiers, qui implique une complexité de sous-mots trop élevée pour permettre à la suite caractéristique des nombres premiers d’être une suite automatique [2].

Ainsi, la récurrence de Giard, bien qu’elle donne un algorithme pour calculer $\sigma(n)$ et donc bien qu’elle fournisse le caractère de primalité de n pour chaque n fixé, **ne fournit pas** un automate fini qui reconnaîtrait la suite caractéristique des primalités. L’automate aurait besoin d’une mémoire qui croît avec n , ce qui est interdit par définition.

7. Conclusion

Notre échange a mis en lumière une distinction essentielle :

- la calculabilité (au sens de Church-Turing) permet des algorithmes dont la mémoire peut croître avec la taille de l’entrée. La formule de Giard en est un exemple.
- la reconnaissance par automate fini impose une mémoire bornée, indépendante de la longueur de l’entrée. Le théorème de Cobham montre que les suites automatiques sont très contraintes.

Votre tentative d’utiliser un automate fini pour “insuffler” la suite de primalité dans votre treillis butait sur cette limitation : l’automate ne peut pas porter en mémoire tous les $\sigma(k)$ précédents pour appliquer la récurrence à chaque nouvelle colonne.