

Proposition de stage de Master 2

Code de Gray: délai et espace constant

1. Contexte et présentation du sujet

Le stage s'inscrit dans le domaine de l'**énumération** et de la **théorie des codes de Gray** avec une attention particulière à la complexité en **espace**. Il pourra éventuellement continuer sur une thèse.

Le problème étudié consiste à produire efficacement toutes les chaînes binaires de longueur donnée ℓ , c'est-à-dire tous les mots de $\{0, 1\}^\ell$, sans répétition et en minimisant le temps écoulé entre la génération de deux mots successifs, appelé *délai* de l'algorithme.

Les algorithmes classiques d'énumération doivent réécrire chaque mot en entier à chaque étape, ce qui impose un délai au moins linéaire en ℓ . Pour dépasser cette limite, la littérature sur les *algorithmes d'énumération* adopte le principe selon lequel la production de la sortie n'est pas comptée dans le temps d'exécution. Ce cadre correspond aux *codes de Gray combinatoires*, où chaque mot est obtenu à partir du précédent par une petite modification (souvent le changement d'un seul bit). Le code de Gray réfléchi, connu depuis plus de 150 ans, a cette propriété, et de très nombreux codes de Gray sont connus [3].

Dans un travail récent [1], nous avons conçu des algorithmes capables d'énumérer l'ensemble $\{0, 1\}^\ell$ avec un **délai constant** et en utilisant seulement une **mémoire auxiliaire constante**, indépendamment de ℓ . Pour modéliser la mémoire constante, on utilise généralement des modèles d'automate. Dans notre cadre d'énumération, nous avons introduit deux modèles computationnels inspirés des machines de Turing : les *tape machines* et les *deque machines*.

- a. Dans le modèle des **tape machines**, le mot courant est stocké sur une bande binaire de longueur ℓ , et cette longueur ne peut pas être modifiée. La tête de lecture se déplace pour modifier un bit à la fois. Nous avons montré [1] qu'il existe une tape machine qui énumère toutes les chaînes binaires de longueur ℓ avec un délai constant entre deux mots successifs, tout en ne modifiant qu'un seul bit à chaque étape.
- b. Dans le modèle des **deque machines**, le mot est stocké dans une file double (deque) dans laquelle on peut ajouter (push) ou retirer (pop) des bits aux deux extrémités. Ce modèle, plus symétrique et ne dépendant pas de la position d'une tête de lecture, permet également d'énumérer toutes les chaînes binaires de longueur ℓ avec un délai constant [1].

2. Objectifs du stage

L'objectif de ce stage est d'abord de découvrir le domaine des codes de Gray, la théorie de la complexité énumérative [5] et les algorithmes d'énumération à délai et espace constant.

On pourra dans un premier temps revisiter les algorithmes pour les tape et deque machines afin d'améliorer leurs paramètres (nombre d'états, délai) ou de les rendre plus simples. En particulier, une manière d'unifier les algorithmes sur les deux modèles de calcul serait intéressant. On pourra aussi enrichir les codes existants d'opérations supplémentaires comme decrement, rank et unrank afin d'en faire des structure de données plus riches (compteurs). Si le candidat est intéressé, l'implémentation existante <https://osf.io/5cg68/> pourra être améliorée et étendue.

Un code k -skew-tolerant, est un code de Gray où entre deux mots consécutifs les bits changés sont à distance au plus k . Le code de Gray implémenté par une tape machine (item a. plus haut) est 3-skew tolerant. Une construction récente [4] produit un code qui est 2-skew-tolerant, mais celui-ci est défini inductivement et n'a pas d'algorithme d'énumération efficace. Nous pourrions essayer de trouver un code de Gray, inspiré de cette construction, qui soit énumérable par une tape machine et 2-skew-tolerant.

Enfin, nous voudrions trouver d'autres langages qu'on peut énumérer avec un délai et un espace constant. Nous voulons d'abord investiguer les codes de Gray dans une base $B > 2$, c'est à dire générer tous les mots de $\{0, \dots, B-1\}^\ell$. L'objectif est ensuite de trouver quels langages réguliers sont énumérables à délai constant par nos modèles de calcul, certaines conditions nécessaires sont déjà connues [2].

3. Compétences requises

Pour mener à bien ce stage, des compétences et surtout un goût pour l'algorithmique, la complexité, les langages formels et les modèles de calculs sont un plus. Néanmoins, ce sujet demande peu de prérequis, les connaissances nécessaires peuvent s'acquérir rapidement.

4. Encadrement

- Le stagiaire sera encadré par Victor Marsault (LIGM) et Yann Strozecki (UVSQ) et collaborera également avec Claire David (LIGM), Nadime Francis (LIGM).
- Le stage aura lieu au Laboratoire d'informatique Gaspard Monge, à l'université Gustave Eiffel (proche de Paris, station Noisy-Champs sur le RER A). Possibilité de travailler aussi au laboratoire DAVID de l'université de Versailles Saint-Quentin-en-Yvelines (UVSQ).
- Les encadrants travailleront avec le stagiaire plusieurs fois par semaine.

Références

- [1] Antoine Amarilli, Claire David, Madime Francis, Victor Marsault, Mikaël Monet, and Yann Strozecki. Gray codes with constant delay and constant auxiliary space. Preprint available

at : http://victor.marsault.xyz/resources/articles/ConstantDelaySpaceGrayCode_v0.pdf, 2025.

- [2] Antoine Amarilli and Mikaël Monet. Enumerating regular languages with bounded delay. In *STACS*, 2023. Preprint available at : <https://arxiv.org/abs/2209.14878>.
- [3] Donald E Knuth. Combinatorial algorithms, part 1, volume 4a of the art of computer programming, 2011.
- [4] Gabriel Sac Himelfarb and Moshe Schwartz. Improved constructions of skew-tolerant Gray codes. *IEEE Transactions on Information Theory*, 71(10), 2025. Preprint available at : <https://arxiv.org/abs/2411.08233>.
- [5] Yann Strozecki. *Enumeration Complexity : Incremental Time, Delay and Space*. Habilitation à diriger des recherches, Université Versailles Saint-Quentin, 2021.