

## Résumé de l'article “Dynamic Membership for Regular Languages”

Cet article a été publié en 2021 dans les actes de la conférence européenne “International Colloquium on Automata, Languages and Programming”, une des conférences de référence en informatique théorique, dont les actes sont disponibles en libre accès. Il a reçu le prix du meilleur article pour la “track B” de cette conférence en 2021 : la notion de “track B” est l’un des deux sous-domaines thématiques de la conférence.

Nous nous intéressons ici au problème de la maintenance incrémentale de l’appartenance à un langage régulier de mots. On fixe un langage cible  $L$ , qui décrit une propriété qui nous intéresse sur les mots. Étant donné en entrée un mot  $w$ , on peut déterminer en temps linéaire en  $w$  si  $w$  appartient ou non au langage  $L$ , par exemple en utilisant un automate fini pour  $L$ . Mais on suppose à présent que  $w$  est modifié par des mises à jour de substitution, par exemple changer le  $i$ -ème caractère du mot pour le remplacer par une lettre  $a$ . Nous cherchons à déterminer de façon efficace, après chaque modification de ce type, si le mot résultant appartient ou non au langage  $L$ . On s’intéresse à la complexité de cette tâche en fonction de la longueur  $n$  du mot  $w$ , qui n’est jamais modifiée par les mises à jour, et on cherche à faire mieux que l’algorithme naïf en  $O(n)$  qui revérifie tout le mot après chaque mise à jour.

Nous démontrons que ce problème peut être résolu en temps logarithmique en  $n$ , plus précisément avec une complexité de  $O(\log n / \log \log n)$  après chaque mise à jour, pour tout langage régulier. Mais cette complexité n’est pas optimale en général, car pour certains langages la maintenance incrémentale est possible en temps constant ( $O(1)$ ), par exemple pour les langages finis. Nous identifions ainsi une classe QLZG de langages pour lesquels la maintenance incrémentale est possible en temps constant, qui couvre notamment les langages finis et les langages commutatifs. Nous démontrons que cette classe est maximale en un sens conditionnel : pour tout langage qui n’est pas dans QLZG, nous démontrons qu’un algorithme de maintenance incrémentale en temps constant pourrait être utilisé pour implémenter une sorte de file de priorité en temps constant, et proposons l’hypothèse que ceci n’est pas possible dans le modèle dans lequel nous nous plaçons. Nous identifions ensuite une classe de langages QSG pour laquelle nous développons un algorithme de maintenance incrémentale en temps  $O(\log \log n)$ , ce qui généralise un résultat antérieur sur les langages apériodiques. Nous montrons enfin que, pour tous les langages réguliers hors de QSG, une borne inférieure inconditionnelle garantit que la maintenance incrémentale est en complexité  $\Theta(\log n / \log \log n)$ .

Ce travail s’inscrit dans une collaboration avec Charles Paperman (Université de Lille), qui a apporté son expertise sur les approches algébriques de la théorie des automates ; et Louis Jachiet (Télécom Paris), qui a proposé des structures de données efficaces et des bornes inférieures sur le modèle abstrait de calcul que nous utilisons. J’ai contribué à la formalisation du problème au cours de

discussions avec Charles, à la conception de certains arguments techniques, à la formalisation de certains résultats algébriques, et j'ai assuré la majeure partie de la rédaction de l'article.