Proposition de thèse : Énumération efficace par éditions successives

Antoine Amarilli, Mikaël Monet

Le domaine des *algorithmes d'énumération* [Was16] étudie des problèmes ayant un grand nombre de solutions et recherche des méthodes pour calculer efficacement toutes les solutions lorsque celle-ci sont nombreuses. Par exemple, on peut chercher des algorithmes efficaces pour énumérer tous les chemins dans un graphe reliant une source et une cible, énumérer tous les mots d'un langage régulier en théorie des langages formels, énumérer toutes les valuations satisfaisantes d'une formule logique, énumérer tous les résultats d'une requête sur une base de données, etc.

Les algorithmes d'énumération cherchent typiquement à garantir des bornes supérieures sur le *délai* entre chaque solution et la suivante. Des algorithmes d'énumération avec un faible délai ont été produits dans de nombreux contextes, et parfois ces algorithmes garantissent même le délai le plus faible possible : un délai *constant* [Seg15] si les résultats à énumérer sont supposés être de taille constante, ou un délai *output-linear* [Bag06] si on ne fait pas cette hypothèse.

Cependant, lorsque les résultats à produire ont une taille non constante, et en particulier quand ils sont grand, alors une garantie *output-linear* sur le délai peut déjà s'avérer trop coûteux. Il semble pourtant impossible d'obtenir une meilleure complexité, à cause du temps nécessaire pour écrire les résultats. L'objectif de ce sujet de thèse est d'explorer une idée générale pour résoudre ce problème : l'*énumération par éditions successives*. Le sujet consistera notamment à étudier ses applications à des cas d'utilisation spécifiques, et ses connexions avec des domaines voisins. La proposition s'appuie sur un premier article des encadrants [AM23] qui a démontré la méthodologie sur une tâche spécifique en théorie des langages formels.

L'idée de l'énumération par éditions consiste à éviter le coût d'affichage des solutions par une simple modification de la définition du modèle : produire chaque solution en *éditant* la solution précédente. Par exemple, lors de l'énumération de chemins dans un graphe, on peut produire chaque chemin en éditant le précédent ; dans l'énumération de mots en théorie des langages formels, on peut appliquer des modifications sur chaque mot pour obtenir le mot suivant. Plus précisément, dans le modèle d'énumération proposé, on peut à tout instant modifier l'état de la mémoire, ou émettre une instruction « sortie » qui renvoie instantanément une partie de l'état courant de la mémoire comme solution, et ce gratuitement du point de vue du temps d'exécution quelle qu'en soit la longueur.

Le défi devient alors de produire les solutions successives en appliquant un nombre constant de modifications entre deux solutions consécutives et en calculant ces modifications aussi efficacement que possible, même lorsque la taille des solutions produites devient grande. L'énumération par éditions est similaire au principe d'énumération combinatoire « *Do not count the output* » [Rus03, p8], mais elle rend ce principe plus réaliste en imposant que chaque sortie soit produite à partir de la sortie précédente par application d'un petit nombre de changements.

Le doctorat étudiera comment cette approche peut s'appliquer à des problèmes dans différents domaines d'application, notamment :

— *Énumération de solutions de requêtes de chemin régulières dans des graphes.* Dans le cadre des bases de données graphe, les *requêtes de chemin régulier* (RPQ) sont un langage théorique qui permet de rechercher des chemins qui satisfont une contrainte donnée par un langage régulier. Formellement, si l'on fixe un alphabet Σ, une base de données graphe est simplement un graphe orienté G avec des arêtes étiquetées par des lettres de Σ ; une RPQ est spécifiée par une expression régulière e sur Σ ; et nous nous intéressons à l'énumération efficace de tous les chemins de G formant un mot de e. La tâche peut être étudiée sous plusieurs sémantiques, en particulier en recherchant des chemins simples ou non, triés ou non par ordre de longueur croissante, etc. Une première direction de recherche consiste à explorer pour quels langages réguliers et quelles sémantiques il est possible de produire efficacement tous les chemins satisfaisant la contrainte régulière en utilisant la méthodologie d'énumération par éditions successives.
— *Langages réguliers.* Une tâche naturelle d'énumération en théorie des langages formels est de produire, à partir d'une description d'un langage L, la séquence des mots de L (potentiellement infinie), éventuellement dans un ordre spécifique. Cette tâche est celle étudiée dans [AM23], mais de nombreuses questions restent ouvertes après cette première exploration du domaine.
— D'autres contextes sont possibles. Une première direction est celle de l'énumération des valuations satisfaisantes pour des formules propositionnelles ou des *circuits booléens*, notamment ceux respectant les restrictions étudiées dans le domaine de la *compilation de connaissances* [DM02]. On pourra également étudier l'énumération des réponses aux *requêtes sur bases de données*, éventuellement dans des ordres spécifiques, sujet qui a été abondamment exploré ces dernières années par la communauté de recherche en théorie des bases de données.

Plus largement, la thèse pourra examiner les liens avec les domaines suivants :

— Les *représentations factorisées*[OZ15]. En effet, l'énumération par éditions revient à identifier les parties communes des solutions : on peut donc potentiellement la rattacher au calcul d'un ensemble factorisé de résultats.
— La *maintenance incrémentale*[AJP21], qui étudie comment déterminer efficacement certaines propriétés des données (par exemple, la connexité d'un graphe, l'appartenance d'un mot à un langage, etc.) et maintenir ces propriétés tandis que les données sont mises à jour par des opérations d'édition.

Objectifs, résultats, programme de travail, livrables et échéancier. Il s'agit d'une proposition de thèse en recherche théorique, ainsi les résultats escomptés sont la définition de problèmes et l'obtention de résultats théoriques sur ces problèmes sous la forme d'algorithmes, de bornes de complexité, etc. Potentiellement, si le candidat ou la candidate s'y prête, une étude expérimentale ou le développement de prototypes logiciels pourront être envisagés. Les résultats de recherche obtenus auront pour vocation à être publiés dans des revues et colloques internationaux du domaine, par exemple les congrès ICDT ou PODS en théorie des bases de données, ICALP ou STACS ou MFCS en informatique théorique plus généralement, ou IJCAI ou AAAI ou NeurIPS en intelligence artificielle.

Le programme global de travail prévu est le suivant, qui devra nécessairement être ajusté par rapport aux spécificités de la recherche théorique :

— 6 premiers mois : étude de l'état de l'art et des prérequis techniques par l'étudiant ou l'étudiante, acquisition des définitions fondamentales, et formalisation d'un problème concret pour l'obtention de premiers résultats
— 12 mois suivants : investigation du problème convenu, obtention de résultats, rédaction des résultats en vue de la soumission d'un premier article
— 18 mois suivants : élargissement de l'étude à d'autres domaines d'application et à d'autres définitions ou variantes du problème posé, rédaction d'autres travaux, présentation des résultats en colloque ou en séminaire ; rédaction du manuscrit de thèse.

Positionnement scientifique et économique. Ce sujet se situe dans le contexte plus large de l'étude des algorithmes d'énumération, notamment dans le domaine de la théorie des bases de données. Ce travail a ainsi de potentielles retombées en termes de résultats théoriques en premier lieu, dans ces domaines ainsi que dans d'autres domaines théoriques à l'interface. Cependant, la recherche en bases de données interagit également avec une communauté de recherche plus appliquée en ingénierie des bases de données ; le sujet proposé a ainsi des retombées potentielles pour le développement de moteurs performants pour la gestion et l'interrogation de grands volumes de données, notamment structurées sous forme de base de données graphe. Le sujet peut également être relié à l'intelligence artificielle via la compilation de connaissances et l'énumération de solutions pour des formalismes de circuits booléens. Ceux-ci peuvent en effet représenter des solutions à des problèmes de raisonnement ou d'optimisation, voire des explications permettant de justifier les décisions prises par les modèles d'intelligence artificielle.

Encadrement et environnement. Cette thèse se déroulera dans l'équipe LINKS du laboratoire CRIStAL de l'Université de Lille, et sera localisée dans le centre de recherche Inria Lille. L'équipe LINKS est spécialisée en logique, algorithmes, théorie des langages formels, et théorie des bases de données, ainsi le sujet s'inscrit directement dans la lignée des thématiques étudiées par l'équipe et des domaines d'expertises de ses membres. La thèse sera co-encadrée par Antoine Amarilli [1] (*advanced research position* à Inria) et Mikaël Monet [2] (Chargé de recherche Inria).

Des collaborations sur ces sujets sont possibles avec d'autres chercheurs de l'équipe LINKS, avec des chercheurs de laboratoires proches : c'est notamment le cas du CRIL à Lens ou le LIGM à Marne-la-Vallée, avec lesquels les encadrants ont déjà des contacts sur des sujets proches de la proposition. Les encadrants ont également un réseau de collaboration international sur des sujets de théorie des bases de données et d'énumération, auquel le doctorant ou la doctorante ont vocation à être intégrés. Ainsi, des collaborations internationales, ou des visites de recherche sous la forme de stages doctoraux par exemple, sont envisageables si cela s'avère pertinent au vu des directions explorées.

Les candidatures doivent être envoyées par email à : `a3nm@a3nm.net` et `mikael.monet@inria.fr`.

Références

[AJP21] Antoine Amarilli, Louis Jachiet, and Charles Paperman. Dynamic membership for regular languages. In *ICALP*, 2021.

[AM23] Antoine Amarilli and Mikaël Monet. Enumerating regular languages with bounded delay. In *STACS*, 2023.

[Bag06] Guillaume Bagan. MSO queries on tree decomposable structures are computable with linear delay. In *CSL*, 2006.

[DM02] Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *JAIR*, 17, 2002.

[OZ15] Dan Olteanu and Jakub Závodnỳ. Size bounds for factorised representations of query results. *TODS*, 40(1), 2015.

[Rus03] Frank Ruskey. Combinatorial generation. Preliminary working draft, 2003.

[Seg15] Luc Segoufin. Constant delay enumeration for conjunctive queries. *ACM SIGMOD Record*, 44(1), 2015.

[Was16] Kunihiro Wasa. Enumeration of enumeration algorithms, 2016.

1. `https://a3nm.net/`
2. `https://mikael-monet.net/`