

Proposition de stage de M2

Reconnaissance de programmes par réseaux de neurones de graphes

Encadrants: Christophe Alias (Inria Lyon, ENS de Lyon)
Sid Touati (Université Côte d'Azur, Sophia-Antipolis),
Enrico Formenti (Université Côte d'Azur, Sophia-Antipolis)
contact: `Christophe.Alias@inria.fr`.

Lieu: ENS de Lyon ou Université de Nice Sophia-Antipolis.

Gratification: \approx 500 euros/mois

Possibilité de poursuite en thèse

Motivations et contexte

La *reconnaissance de programmes* est un problème fondamental en informatique, qui trouve de nombreuses applications (optimisation de programmes, décompilation, génie logiciel, etc). En général, le problème de l'équivalence de calcul entre deux programmes est indécidable. Il existe plusieurs stratégies pour contourner cette difficulté. On peut ne considérer qu'un sous-ensemble de programmes, dont le contrôle (les boucles notamment) est prédictible statiquement [1, 2, 4] ou encore restreindre la relation d'équivalence avec des approches plus syntaxiques [5, 3].

Dans ce stage, on se propose de relâcher la relation d'équivalence en *acceptant de faire des erreurs de temps en temps*, tout en veillant à ce que les erreurs soient aussi rares que possible. C'est un cadre d'application idéal des réseaux de neurones qui, par nature, calculent leurs résultats à une marge d'erreur près. Pour ce, nous nous appuierons sur les développements récents autour des *Graph Neural Network* [6].

Présentation du sujet de stage de recherche

Ce stage a pour objectif de réfléchir à une nouvelle démarche de reconnaissance de programmes, qui n'est pas intuitive. L'intelligence artificielle est un domaine classique issu des mathématiques et des sciences informatiques, qui a montré son efficacité en pratique dans des situations de reconnaissance de formes graphiques, de parole, de texte, d'interaction homme-machine, etc. Nous souhaitons développer un outil pour reconnaître des programmes basé sur l'intelligence artificielle. Nous appliquerons des méthodes de réseaux de neurones pour les classification des graphes (Graph Neural Network). Les étapes de travail sont:

1. Modifier le code source d'un compilateur existant (LLVM) pour extraire le graphe de flot de données d'un programme sous un format texte approprié.
2. Générer via compilation plusieurs variantes d'un programme, et entraîner un réseau de neurones avec la méthode GNN (Graph Neural Network).

3. Tester si et à quel point un tel réseau de neurones peut reconnaître d'autres variantes du même programme.

Prérequis souhaités

Le candidat idéal pour ce stage est un étudiant qui a suivi une formation en informatique fondamentale (compilation, programmation avancée, réseaux de neurones, graphes). Concernant la partie technique de compilation avec LLVM, la maîtrise du C++ est nécessaire.

References

- [1] Christophe Alias and Denis Barthou. Algorithm recognition based on demand-driven data-flow analysis. In *IEEE Working Conference on Reverse Engineering (WCRE'03)*, 2003.
- [2] Guillaume Iooss, Christophe Alias, and Sanjay Rajopadhye. On program equivalence with reductions. In *21st International Static Analysis Symposium (SAS'14)*, Munich, Germany, September 2014.
- [3] Shlomit S. Pinter and Ron Y. Pinter. Program optimization and parallelization using idioms. In *Proceedings of the 18th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL'91, pages 79–92, New York, NY, USA, 1991. ACM.
- [4] Sven Verdoolaege, Gerda Janssens, and Maurice Bruynooghe. Equivalence checking of static affine programs using widening to handle recurrences. *ACM Transactions on Programming Languages and Systems*, 34(3):1–35, November 2012.
- [5] Linda M. Wills. *Automated Program Recognition by Graph Parsing*. PhD thesis, MIT, July 1992.
- [6] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.