Étendre le Machine Learning Symbolique pour l'Analyse de Code

Mots clés: machine learning symbolique, analyse de binaires, annotations, reverse engineering

Encadrant Principal
Grégoire MENGUY
CEA LIST
gregoire.menguy@cea.fr

Superviseur Sénior
Sébastien BARDIN
CEA LIST
sebastien.bardin@cea.fr

Les annotations, comme les contrats de fonction, les invariants de boucle ou les types, permettent de spécifier un code. Celles-ci sont donc cruciales pour améliorer la sécurité des programmes et réaliser différentes tâches, de l'ingénierie logicielle à la vérification. Malheureusement, ses annotations (e.g., contrats de fonction et invariants de boucle) sont rarement fournies et celles qui le sont (e.g., les types) sont usuellement perdue à la compilation. Ils doivent donc être rétro-ingéniés à la main.

Récemment, l'algorithme PreCA [5] a été proposé pour inférer automatiquement, via du machine learning symbolique [1], des contrats de fonctions et avec des garanties fortes de correction. Actuellement, PreCA a été utilisé sur des code binaires pour inférer des propriétés mémoires comme la validité ou l'aliasing de pointeurs. Néanmoins, il s'agit d'un framework général pouvant être appliqué sur de nombreux autres cas. Une extension naturelle serait donc de permettre à PreCA d'inférer des préconditions plus avancés comme des annotation SAL (Microsoft) [3] ou E-ACSL (Frama-C) [4]. Un autre axe pourrait être de généraliser PreCA pour synthétiser des contrats hardware/software [2] (utilisés pour lutter contre les attaques Spectre), ou d'autres relations comme des frame conditions.

Le projet se déroulera comme suit:

- Comprendre et prendre en main l'outil d'inférence de précondition PreCA;
- Étudier et sélectionner un axe d'extension de PreCA;
- Implémenter cette extension dans le projet open source PreCA;
- Évaluer cette extension pour prouver son efficacité en pratique.

1 Livrables attendus

Les principaux livrables attendus sont:

- Un résumé des recherches bibliographiques menées;
- Une implémentation documentée de l'extension proposée;
- Le rapport final (rédigé ou slides) incluant les deux premiers livrables et le récapitulatif des démarches suivis et des résultats obtenus.

2 Éléments logistiques

Des points réguliers seront organisés avec l'encadrement. Ils pourront être réalisés en visioconférence ou au CEA sur le site Nano-Innov à Saclay.

3 Postuler

Pour postuler, les étudiants doivent contacter l'encadrant par mail (avec le superviseur sénior en CC).

References

[1] Christian Bessiere et al. "Constraint acquisition". In: *Artificial Intelligence Journal* 244 (2017), pp. 315–342. DOI: 10.1016/j.artint.2015.08.001. URL: https://doi.org/10.1016/j.artint.2015.08.001.

[2] Marco Guarnieri et al. "Hardware-software contracts for secure speculation". In: *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2021, pp. 1868–1883.

[3] Michael Howard. *A brief introduction to the standard annotation language (sal)*. 2006.

[4] Florent Kirchner et al. "Frama-C: A software analysis perspective". In: *Formal Aspects Comput.* 27.3 (2015), pp. 573–609. DOI: 10.1007/s00165-014-0326-7. URL: https://doi.org/10.1007/s00165-014-0326-7.

[5] Grégoire Menguy et al. "Automated Program Analysis: Revisiting Precondition Inference through Constraint Acquisition". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. Ed. by Lud De Raedt. Main Track. International Joint Conferences on Artificial Intelligence Organization, July 2022, pp. 1873–1879. DOI: 10.24963/ijcai.2022/260. URL: https://doi.org/10.24963/ijcai.2022/260.