

Learning Exact Loop Bound Through Programming-by-Example

Keywords: Program analysis, artificial intelligence, program synthesis

<i>Main Supervisor</i>	<i>Senior Supervisor</i>
Grégoire MENGUY	Sébastien BARDIN
<i>CEA LIST</i>	<i>CEA LIST</i>
<i>gregoire.menguy@cea.fr</i>	<i>sebastien.bardin@cea.fr</i>

Determining loop bounds, i.e., the number of loop iterations w.r.t., the provided inputs, is critical in program analysis. Many methods are limited to recovering an approximate bound. Unfortunately, it is not precise enough in many contexts. Recovering the exact loop bound is much harder but also much more useful. It helps with code verification of critical and real-time systems [4], enables comparison of software execution costs, and even assesses the conditional equivalence of execution costs [7].

Nevertheless, exact loop-bound synthesis is still in its infancy. Riley, Daniel et al. [7], recently proposed ELBA, the first framework to tackle this problem. By combining Integer Linear Programming [6] and verification [8], ELBA can efficiently find the loop bound and prove its correctness. Unfortunately, ELBA is by design limited to simple loops (no nested loops, with linear conditions only). How to design general and efficient exact loop-bound synthesizers remains an open problem.

We believe that programming-by-example [3] is a strong candidate for taking one step further. Indeed, it has shown to be highly efficient in multiple contexts from EXCEL MACRO synthesis [2] to binary code deobfuscation [5, 1]. Especially, the open source XYNTIA synthesizer from the CEA (<https://github.com/binsec/xyntia>) is the state-of-the-art synthesizer for deobfuscation [5]. We recently evaluated XYNTIA for *exact loop-bound synthesis* and provided dedicated extensions. As expected, XYNTIA synthesis highly outperforms ELBA.

In this project, we aim to take one step further by (i) integrating XYNTIA inside the ELBA verification loop; and (ii) evaluating XYNTIA over more challenging benchmarks. It will proceed as follows:

- Get familiar with the state-of-the-art of program synthesis and exact loop bound analysis;
- Integrate XYNTIA into the ELBA verification loop;
- Evaluate XYNTIA over exact loop bound analysis tasks found in the literature.

1 Expected deliverable

- A summary of the bibliography made by the student;
- A documented implementation of the proposed extensions;
- A final report (slides) with the first two deliverables and a summary of the research and results.

2 Organization

Regular meetings will be organized with the supervisor (online or in person at CEA Nano-Innov, Saclay).

3 How to apply

To apply, students must send an email to the main supervisor (Grégoire Menguy) with the senior supervisor (Sébastien Bardin) in copy.

References

- [1] Tim Blazytko et al. “Syntia: Synthesizing the Semantics of Obfuscated Code”. In: *USENIX Security*. 2017.
- [2] José Cambronero, Sumit Gulwani, and Vu Le et al. “FlashFill++: Scaling Programming by Example by Cutting to the Chase”. In: *Proc. ACM Program. Lang.* POPL (2023).
- [3] Sumit Gulwani, Oleksandr Polozov, Rishabh Singh, et al. “Program synthesis”. In: *Foundations and Trends® in Programming Languages* (2017).
- [4] Jane WS Liu et al. *Real-time systems*. Pearson Education India, 2006.
- [5] Grégoire Menguy, Sébastien Bardin, and Bonichon et al. “Search-Based Local Black-Box Deobfuscation: Understand, Improve and Mitigate”. In: *Conference on Computer and Communications Security*. 2021.
- [6] George L Nemhauser and Laurence A Wolsey. *Integer and combinatorial optimization*. Vol. 18. Wiley New York, 1988.
- [7] Daniel Riley and Grigory Fedyukovich. “Exact Loop Bound Analysis”. In: *Proceedings of the ACM on Programming Languages* 9.PLDI (2025), pp. 1814–1837.
- [8] Kurt Sieber. *The foundations of program verification*. Springer-Verlag, 2013.