Backward Abstract Interpretation using Over and Under-Approximations

Master 2 research internship proposal, 2018–2019

Supervisor:	Antoine Miné (`antoine.mine@lip6.fr`)
Internship location:	APR team, LIP6 Sorbonne Université Jussieu Campus, Paris, France
Duration:	4.5 to 6 months
Related project:	Mopsa project
Relevant courses:	– MPRI 2.6: Abstract interpretation: application to verification and static analysis – Master STL: Typage et analyse statique

Other internships are possible on the topic of static analysis and abstract interpretation. Contact the internship supervisor for more information.

The goal of the internship is to develop backward value analyses in a static analyzer by abstract interpretation in a realistic setting for C-like programs featuring notably functions and pointers. The intern will consider both over-approximations and under-approximations.

Backward Static Analysis

The theory of abstract interpretation allows the design of effective and efficient static analyzers able to compute approximations of program semantics. The overwhelming majority of analyzers compute over-approximations:

1. A classic forward analysis computes an over-approximation of the states reachable from the program entry.

2. A classic backward analysis computes the co-reachable states from some desired states. This set is also over-approximated, so as to infer necessary conditions for the program to reach the desired states.

3. Over-approximating forward and backward analyses can be iterated, either to provide automatically some context for the false positives of an imprecise analysis [5], or to provide interactive abstract debugging [3].

In theory, an under-approximating backward analysis could instead provide *sufficient conditions* for a target program state to be reached. Applications include:

1. **Proving that an alarm is a true error** and not a false alarm by inferring a counter-example execution [1].

2. **Inferring procedure contracts**: sufficient assertions to insert at the beginning of a procedure to ensure that its execution will never fail [2].

3. **Evaluating the uncertainty of an analysis** by combining both over-approximations and under-approximations, and quantifying the distance between them. Such analyses may be iterated to improve their precision.

The lack of effective under-approximating infinite-state abstract domains has limited the development of Abstract Interpretation techniques to solve these problems. Counter-example generation with formal verification has been widely explored in the context of model-checking, but requires reasoning in a finite or regular world that can be exactly represented in a symbolic model-checker; the internship will explore instead the analysis of large state spaces that require sound approximations to scale up. Procedure contract inference by Abstract Interpretation has been proposed by Cousot et al. [2], but employs over-approximations of necessary conditions to remove only erroneous executions; the internship will also consider under-approximated sufficient conditions to keep only correct executions.

Effective under-approximations have been proposed for classic numeric domains (intervals, polyhedra) in [1] to infer sufficient conditions for the absence of array bound errors in simple programs, and later found applications in proving liveness properties [4]. These can serve as the basis for the internship, but will require extensions to ensure precision, scalability, and support for non-numeric variables such as pointers.

Expected Work

The intended work will include a theoretical side: developing abstract semantics and proving formally their soundness. It will also include a practical side: implementing the semantics and validating their benefit experimentally.

The host team is developing a static analysis platform, Mopsa, that includes an analysis of C programs using several, ready-to-use abstractions, and a framework to easily extend it to new abstractions. However, the framework is currently limited to forward over-approximating analyses. A first step will be to add support for backward iterations in Mopsa, and evaluate classic over-approximating backward analyses on programs featuring numeric and pointer variables. Then, the intern will implement and evaluate the novel over-approximating and under-approximating backward operators developed during the internship. Feedback from experimentations throughout the internship will guide the design of new abstractions tailored to concrete problems in realistic settings.

Requested Skills

The internship requires a strong knowledge of static analysis by abstract interpretation. The intern should have followed one of the following Master 2 courses: “Abstract interpretation: application to verification and static analysis” from MPRI, “Typage et analyse statique” from the STL Master at Sorbonne Université, or an equivalent course. Knowledge of the OCaml language is also required, for the implementation effort within the Mopsa platform.

Context of the Internship

The internship will take place in the APR team, in the LIP6 laboratory, Jussieu Campus, Sorbonne Université, Paris. It is proposed in the scope of the Mopsa ERC research project. If the internship is successful, the project may provide opportunities for a funded PhD on a follow-up subject.

References

[1] A. Miné. Backward under-approximations in numeric abstract domains to automatically infer sufficient program conditions. *Science of Computer Programming (SCP)*, 33 pages, Oct. 2013. `http://www.di.ens.fr/~mine/publi/article-mine-SCP13.pdf`.

[2] P. Cousot, R. Cousot, & F. Logozzo. Precondition Inference from Intermittent Assertions and Application to Contracts on Collections. In *12th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'11)*, Austin, Texas, LNCS 6538, Springer, 2011, pp. 150–168.

[3] F. Bourdoncle. Assertion-Based Debugging of Imperative Programs by Abstract Interpretation. In Proc. of 4th European Software Engineering Conf. (ESEC'93), pp. 501–516, vol. 717 of LNCS. Springer, 1993.

[4] C. Urban, S. Ueltschi, and P. Müller Abstract Interpretation of CTL Properties. In Proc. of SAS'18, 402–422, Freiburg, Germany. Springer.

[5] X. Rival. Understanding the origin of alarms in Astrée. In Proc. of SAS'05, 303–319. Springer.