INRIA — Département d'Informatique de l'École Normale Supérieure

Abstract Domain Reduction Operation Beween Separation Logic Memory Abstraction And Value Abstraction

Internship location: École Normale Supérieure ; 45, rue d'Ulm ; 75 230, PARIS.

Team: Équipe Sémantique et Interprétation Abstraite / Équipe-Projet "ANTIQUE".

Advisor & Contact : Xavier RIVAL (*e-mail* : `rival@di.ens.fr`, tél : 01 44 32 21 50, fax : 01 44 32 21 51)

Internship topic:

Shape analyses such as [1,2] aim at computing precise structural invariants over memory states. For instance, they can infer properties of linked data-structures, and help verifying safety properties such as absence of memory errors or the preservation of global structural invariants. They use separation logic [3] in order to describe memory states: they describe separate memory regions with logical predicates that either very precisely define the contents or memory cells, or summarize them using high level recursive predicates. Such predicates can describe structures such as linked lists and binary trees.

Memory abstract predicates can be combined with value properties in many ways. For instance, we can pair them with numerical predicates in order to describe complex arborescent structures [2]. We can also combine them with constraints over sets [4], which allows to describe data-structures with some amount of sharing, provided they have at least one recursive backbone (the best example is a graph with a representation based on adjacency lists).

However, this combination requires efficient support for *reduction*, namely the communication of information across domains. For instance, when a shape predicate asserts that a list is non empty, and a numerical predicate asserts that the head pointer to that list is null, the combination of the two predicates describes the empty set of states and should be replaced with $\bot$, which more explicitely and simply describes the same semantic property. Similarly, information about sortedness, cell content ranges and pointer numerical comparisons may be used to refine memory predicates. In the opposite direction, memory predicates may often be used in order to refine value predicates.

Existing works [2,4] feature a partial and ad hoc support for reduction which is not a satisfactory general solution.

The purpose of this internship is to study a more general form of reduction, to implement it into the MemCAD static analysis tool, and to evaluate it. The work plan involves the following steps

1. **Definition and formalization of reduction primitives across shape abstract domains and value abstract domains**. This task consists in identifying adapted forms of logical predicates for the representation of memory properties and of content properties which allow efficient communication across abstract domains. In particular, this step will involve an in-depth study of the form of the inductive logical predicates in the shape abstract domain. For this work, we will consider the abstract domain of [2] as a starting point.
2. **Design of algorithms for reduction**. The second task aims at designing efficient and accurate algorithms to achieve a good level reduction of memory and value predicates. Reduction algorithms should not be too costly and should guarantee a high level of precision of the analyses.
3. **Implementation and evaluation**. The final step consists in the implementation and evaluation of the analysis. This implementation will be done in the context of the **MemCAD** static analyzer [5,6], so as to reuse the definition of basic logical predicates and analysis algorithms, and implement only the new constructions and algorithms defined in the two previous points. Evaluation will focus on complex data-structures found in the Linux kernel.

Pre-requisite:

For this internship, it would be preferable that the student is familiar enough with semantics and abstract interpretation, but a candidate interested in the foundations of programming languages would also be able to acquire the required knowledge.

References

[1] Dino Distefano, Peter W. O'Hearn, Hongseok Yang. A Local Shape Analysis Based on Separation Logic. In TACAS'06, pages 287-302, 2006.

[2] Bor-Yuh Evan Chang and Xavier Rival. Relational inductive shape analysis. In POPL'08, pages 247–260, 2008.

[3] John C. Reynolds. Separation Logic: A Logic for Shared Mutable Data Structures. In LICS'02, pages 55-74, 2002.

[4] Huisong Li, Xavier Rival, Bor-Yuh Evan Chang. Shape Analysis for Unstructured Sharing. In SAS'15, pages 90-108, 2015.

[5] MemCAD static analyzer. https://www.di.ens.fr/~rival/memcad.html

[6] Huisong Li, François Bérenger, Bor-Yuh Evan Chang, and Xavier Rival. Semantic-Directed Clumping of Disjunctive Abstract States. In POPL'17, pages 32-45, 2017.

[7] Patrick Cousot, Radhia Cousot. Systematic Design of Program Analysis Frameworks. In POPL'79, pages 269-282, 1979.