Research Internship – Master M2 (2014/2015)

Location : Laboratoire Spécification et Vérification
ENS de Cachan

Title : Translation methods for deciding separation logics

Advisor : Stéphane Demri (LSV, CNRS, ENS Cachan)
demri@lsv.ens-cachan.fr

Direct approach vs. translation for non-classical logics In order to mechanize non-classical logics, there exist at least two main approaches. The direct approach consists in building specialized proof systems for the logics and requires building new theorem provers but, it has the advantage to design fine-tuned tools and to propose plenty of optimizations. The development of tableaux-based provers for description logics perfectly illustrates this trend. By contrast, the translation approach consists in reducing decision problems for the source logics to similar problems for target logics that have already well-established theorem provers. Its main advantage is to use existing tools and therefore to focus only on the translations, that are usually much simpler to implement.

Separation logic Separation logic has been introduced as an extension of Hoare logic [Hoa69] to verify programs with mutable data structures [IO01, Rey02]. A major feature is to be able to reason locally in a modular way, which can be performed thanks to the separating conjunction that allows to state properties in disjoint parts of the memory. Moreover, the adjunct implication asserts that whenever a fresh heap satisfies a property, its composition with the current heap satisfies another property. This is particularly useful when a piece of code mutates memory locally, and we want to state some property of the entire memory (such as the preservation of data structure invariants). The development of proof methods for separation logic (and its fragments and variants) is nowadays a very active area, see e.g. [GM10, BV14, DGLWM14].

Translation vs. specialized algorithms for separation logic Despite its young age, one can observe that the mechanization of separation logic follows a similar dichotomy. This is all the more obvious nowadays since there are a lot of activities to develop verification methods with decision procedures for fragments of practical use, see e.g. [CHO$^+$11]. Many decision procedures have been designed for fragments of separation logics or abstract variants, from analytic methods [GM10] to translation to theories handled by SMT solvers [PWZ13], passing via graph-based algorithms [HIOP13]. The framework of satisfiability modulo theories (SMT) [BT14] remains probably the most promising one to develop decision procedures dedicated to reasoning tasks for separation logics, see e.g [PWZ13].

SMT solvers, program verification, and separation logic Deciding logical formulae within a given logical theory is ubiquitous in computer science and the works around Satisfiability Modulo Theories (SMT) are dedicated to solve this problem by providing methods, proof systems and solvers in order to be able to decide as much theories as possible, as well as their combination (see e.g. [BT14]). Nowadays, SMT solvers are essential for most tools that formally verify programs. A nice feature of such solvers is their ability to combine distinct theories allowing to express richer statements. As advocated in [PWZ13], being able to integrate decidable fragments of separation logic in some SMT solver not only allows to decide satisfiability or entailment problems by taking advantage of the technology behind SMT solvers but also it provides an efficient way to combine separation logics with other theories. This provides an important step to integrate reasoning about separation logic into SMT solvers.

Objectives of the research internship

(1) To become familiar with tractable [resp. decidable] fragments of separation logic and with existing logical theories for deciding separation logic. A selection of relevant documents shall be provided to the intern before the beginning of the internship.

(2) To analyze the features of existing translations from decision problems for separation logic to decision problems for other logical theories. The goal is to identify the cases when SMT solvers, or other types of provers, can be used at their best.

(3) To propose new translations for extensions of existing fragments; candidate fragments have been already identified in the literature.

(4) If time permits, to implement a prototype of a translation designed during the internship, and to run a prover for the target logical theory on formulae generated from it.

This research internship may be pursued as a PhD thesis, whose subject may vary according to the candidate's research interests.

Related courses at MPRI:

For your information, the following MPRI courses are related to this research internship. Students from other master programmes are welcomed to apply too.

- Course 2.5.1 *Automated Deduction*
- Course 2.9.1 *Mathematical foundations of the theory of infinite transition systems*
- Course 2.9.2 *Algorithmic verification of programs*
- Course 2.36.1 *Proofs of programs*

References

[BT14] C. Barrett and C. Tinelli. satisfiability modulo theories. In *Handbook of Model Checking*. 2014. In preparation.

[BV14] J. Brotherston and J. Villard. Parametric completeness for separation theories. In *POPL'14*, pages 453–464. ACM, 2014.

[CHO$^+$11] B. Cook, C. Haase, J. Ouaknine, M. Parkinson, and J. Worrell. Tractable reasoning in a fragment of separation logic. In *CONCUR'11*, volume 6901 of *Lecture Notes in Computer Science*, pages 235–249. Springer, 2011.

[DD14] S. Demri and M. Deters. expressive completeness of separation logic with two variables and no separating conjunction. In *LICS'14*, page 37. ACM, 2014.

[DGLWM14] S. Demri, D. Galmiche, D. Larchey-Wendling, and D. Mery. separation logic with one quantified variable. In *CSR'14*, volume 8476 of *Lecture Notes in Computer Science*, pages 125–138. Springer, 2014.

[GM10] D. Galmiche and D. Méry. Tableaux and resource graphs for separation logic. *Journal of Logic and Computation*, 20(1):189–231, 2010.

[HIOP13] C. Haase, S. Ishtiaq, J. Ouaknine, and M. Parkinson. Seloger: A Tool for Graph-Based Reasoning in Separation Logic. In *CAV'13*, volume 8044 of *Lecture Notes in Computer Science*, pages 790–795. Springer, 2013.

[Hoa69] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 1969.

[IO01] S. Ishtiaq and P. O'Hearn. BI as an assertion language for mutable data structures. In *POPL'01*, pages 14–26. ACM, 2001.

[PWZ13] R. Piskac, Th. Wies, and D. Zufferey. automating separation logic using smt. In *CAV'13*, volume 8044 of *Lecture Notes in Computer Science*, pages 773–789. Springer, 2013.

[Rey02] J.C. Reynolds. Separation logic: a logic for shared mutable data structures. In *LICS'02*, pages 55–74. IEEE, 2002.