Certification of Distributed Self-Stabilizing Algorithms Using Coq

Karine Altisen Pierre Corbineau Stéphane Devismes

Verimag Lab
Contacts: {Karine.Altisen, Pierre.Corbineau, Stephane.Devismes}@imag.fr

Scientific Context. Modern distributed systems can be *large-scale* (*e.g.*, Internet), *dynamic* (*e.g.*, Peer-to-Peer systems), and / or *resource constrained* (*e.g.*, wireless sensor networks – WSNs). Those characteristics increase the number of faults which may hit the system. For instance, in WSNs, processes are subject to crash failures because of their limited battery. Moreover, their communications use radio channels which are subject to intermittent loss of messages. Now, due to their large-scale and the adversarial environment where they may be deployed, intervention to repair them cannot be always envisioned. In this context, fault-tolerance, *i.e.*, the ability of a distributed algorithm to endure the faults by itself, is mandatory.

Self-stabilization [3] is a versatile lightweight technique to withstand transient faults in a distributed system. After transient faults hit and place the system into some arbitrary global state, a self-stabilizing algorithm returns, in finite time, to a correct behavior without external intervention. Self-stabilization makes no hypotheses on the nature or extent of transient faults that could hit the system, and recovers from the effects of those faults in a unified manner.

Today's researches on self-stabilizing algorithms focus on more and more complex problems and adversarial environments. This makes the proof that an algorithm actually achieves self-stabilization even more complex and subtle to establish. Now, those proofs are usually performed by hand, using informal reasoning. Such methods are clearly pushed to their limits and this justifies the use of a *proof assistant*.

Proof assistants are environments in which a user can express programs, state theorems, and develop proofs interactively, those ones being mechanically checked (*i.e.*, machine-checked). In particular, the Coq proof assistant [4], which is targeted by this project, has been successfully used for various tasks such as mathematical developments as involved as the 4-colors or Feit-Thompson theorems, formalization of programming language semantics leading to the certification of a C compiler, certified numerical libraries, and verification of cryptographic protocols.

Project. We propose a *framework, based on* Coq*, to (semi-) automatically construct certified proofs of self-stabilizing algorithms*. The framework is currently under development [2] and a first experiment has been conducted, with the certification of a non-trivial case study [1]. This work imports into Coq the computational model in which the targeted algorithm is designed, formalizes the algorithm itself and its specification. Then the algorithm is proved using Coq including safety, convergence and also some performance analyses.

In this internship, we propose to contribute to further developments of the framework. As an example, the design of self-stabilizing algorithms is usually compositional: two simple algorithms can be used together (composed) to create a more complex one. We aim at further developing Coq libraries about composition of algorithms and to validate them under case studies. Precisely, the subject requires:

- A bibliographical study and a deep understanding of the framework [2], [1]. This may require learning some Coq skills, if necessary.
- To develop further the existing Coq libraries under one of the following directions: composition of algorithms and tools to prove their safety and convergence; tools for proving performance results of algorithms; development and application to the case study.

Required Skills. An important background about sequential algorithmic, in particular proof of algorithms, is mandatory. Background about distributed systems and formal methods is a plus.

Working context. The internship is part of ANR project ESTATE[1]. The student will be integrated in the lab Verimag[2].

Possible extension into a PhD thesis.

2017-2018

References

[1] Karine Altisen, Pierre Corbineau, and Stéphane Devismes. A framework for certified self-stabilization. *Logical Methods in Computer Science (special issue of FORTE 2016)*, 2017. To appear.

[2] Karine Altisen, Corbineau Pierre, and Stéphane Devismes. Padec: A framework for certified self-stabilization, 2016. http://www-verimag.imag.fr/~altisen/PADEC/.

[3] E. W. Dijkstra. Self-Stabilizing Systems in Spite of Distributed Control. *Commun. ACM*, 17:643–644, 1974.

[4] The Coq Development Team. *The Coq Proof Assistant Documentation*, June 2012. http://coq.inria.fr/refman/.

[1] https://wp-systeme.lip6.fr/estate/
[2] http://www-verimag.imag.fr/