

INTERNSHIP SUBJECT PROPOSAL

Research teams name: AlGorille and Mosel

Research Unit: Nancy – Grand Est

Research theme: NUM (AlGorille) and SYM (Mosel)

Research team leader: Jens Gustedt (AlGorille) and Dominique Méry (Mosel)

Intern tutors: Martin Quinson (AlGorille) and Stephan Merz (Mosel)

Intern level: Master or PhD student

Internship duration: 4 to 6 months

Possibility of a follow-up Ph-D: yes

Model-Checking Distributed Algorithms with PlusCal 2.0

Distributed algorithms are known for being tedious to design and assess. Since they are based on the interaction of several independent components, they are hard to envision for the sequential human brains. Deadlocks, race conditions and resource starvation are unfortunately common problems of distributed computing.

Model checking is one of the most successful techniques to assess that a distributed algorithm exhibits the expected properties. The main idea is to exhaustively build and analyse the state space of an algorithm and check that the requirements are satisfied for all possible executions. For a distributed algorithm, all possible interleavings of the different processes will be analyzed.

Usually, the users of such methods are expected to express a formalization of their algorithm using a mathematical formalism such as TLA⁺ [1]. In order to remain usable by non-specialist and also to reduce the possibility of bug introduction during the manual translation between the mathematical formalism of TLA⁺ and a more programming oriented one, Lamport introduced the PlusCal [2] language. The algorithms are expressed in a natural manner and then converted automatically to TLA⁺ using a dedicated compiler. We recently extended PlusCal to a 2.0 version easing the expression of distributed algorithms in addition to the concurrent algorithms targeted by the original version.

1 Internship description

The goal of the proposed internship is to assess the effectiveness of this approach and the proposed tool by writing several distributed algorithms in the PlusCal 2.0 formalism. First, we aim at expressing in PlusCal 2.0 several classical algorithms such as leader election (Lamport's algorithm, bully algorithm, etc.), mutual exclusion (either quorum based or token based), wave propagation. These should constitute a set of ready to use examples easing the learning curve of the proposed formalism. Then, we plan to express several algorithms recently published in renowned conferences. This will hopefully demonstrate the effectiveness of the approach by allowing to identify glitches and modeling inaccuracies in published work.

References

- [1] L. Lamport. *Specifying Systems*. Addison-Wesley, Boston, Mass., 2002. See also <http://research.microsoft.com/users/lamport/tla/tla.html>.
- [2] L. Lamport. *Checking a Multithreaded Algorithm with +CAL*. Distributed Computing (DISC 2006). Lecture Notes in Computer Science. See also <http://research.microsoft.com/users/lamport/tla/pluscal.html>.