Internship proposal
Verification of Distributed Algorithms

Advisors: Anca Muscholl, Igor Walukiewicz, Corentin Travers

Laboratory: LaBRI, Laboratoire Bordelais de Recherche en Informatiqye. 351 cours de la libération, 33400 Talence.

Duration: 4 to 6 months, from March 2018

Funding: Standard internship gratification funded by the ANR project FREDDA

Contact: Corentin Travers `travers@labri.fr`

Context: Distributed applications represent a significant part of our everyday life. Typically, our personal data are stored on remote distributed servers, data management relies on remote applications reachable via smartphones, data-intensive computations are performed on computer clusters, etc. Since distributed applications are deployed at increasingly large scale, they have to be reliable and robust, satisfying stringent correctness criteria.

Distributed algorithms, are most often tailored for a specific system model. For example, communication can be synchronous or asynchronous, or a certain number of errors and failures of certain type can happen during executions. These assumptions are crucial and may lead to different families of algorithms designed for a specific context. While there exists a large collection of sophisticated techniques for algorithm design and analysis, most of the existing distributed algorithms, together with their correctness proofs, remain tailored to one very specific framework and are surprisingly difficult to generalize to different, but apparently similar frameworks.

In the context of distributed computing, formal methods aim at automatically asserting correctness of distributed algorithms. When the algorithm under consideration is designed for a fixed finite number of processes using only finite-domain variables, software tools such as SPIN [2] are able to show correctness properties automatically. In general, however, distributed algorithms are designed for a non-fixed number of processes and may use variables whose domain are not bounded. *Parameterized verification* amounts to establishing correctness of an algorithm no matter how many processes actually interact. A widely used proof technique in parameterized verification is the *cut-off principle* [1], which builds on the following deduction rule: If the algorithm is correct when at most N processes intervene, then it is correct for any number of processes. The correctness of the algorithm then boils down to verify a finite-state system consisting in N processes.

Goals: The main goal of this internship is to develop new verification techniques for distributed algorithms. The work will focus on algorithms for *agreement tasks*, such as *consensus*. A good starting point might be trying to remove some of (the many) restrictions in [3] or/and extend the cutoff bounds to more severe failures models, such as byzantine failures.

References

[1] E. A. Emerson and K. S. Namjoshi. On reasoning about rings. *Int. J. Found. Comput. Sci.*, 14(4):527550, 2003.

[2] Gerard J. Holzmann. Software model checking with SPIN. *Advances in Computers*, 65:78109, 2005.

[3] Marić O., Sprenger C., Basin D. Cutoff Bounds for Consensus Algorithms. *Computer Aided Verification (CAV'17)*. Lecture Notes in Computer Science, vol 10427. Springer, 2017