Formal Mechanical Models for Distributed Computing - Validating Model Checker Input

Location: LIRIS
Bât Nautibus, Université Claude Bernard Lyon 1
69100 Villeurbanne

Supervision:
Xavier Urbain (Lyon), Pierre Courtieu (Paris)
Tel.: 01 40 27 24 13, 04 27 46 57 07,
Email: xavier.urbain@univ-lyon1.fr, pierre.courtieu@cnam.fr

Head of Laboratory:
Jean-Marc Petit
Email: jean-marc.petit@insa-lyon.fr

Context and Scientific Goals

This work takes place within the ANR project SAPPORO (2020-2024) involving Université Lyon-1, Sorbonne Université, CNAM Paris (France), and Tokyo Institute of Technology (Japan).

https://sapporo.liris.cnrs.fr

Distributed computing is one of the domains where informal reasoning is not an option, in particular when Byzantine failures are involved. What characterises also Distributed Computing is its diversity of models subtle modifications of which induce radical change in the system behaviour. We consider Robot Networks, that is swarms of autonomous mobile entities that have to accomplish some task in cooperation. In this emerging framework, models can be distinguished by the capabilities of robots, the topology of the considered space, the level of synchrony (that is the properties of a demon), the type of the failures likely to occur, etc.

We are interested in obtaining formal and moreover mechanical guarantees of properties for certain protocols, using the Coq proof assistant. A Coq framework\(^1\) for robot networks recently proposed can express quite a few variants of models for such networks, in particular regarding topology or demon properties. This framework is higher order and based on coinductive types. It allows to prove in Coq positive results (the task will be fulfilled using the algorithm embedded in all robots for all initial configuration) \([4, 2]\) as well as negative results (there cannot be any embedded algorithm that will work for this task for all initial configuration) \([3, 1]\).

This internship focuses on consolidating the model checking approach to verification of protocols by proving the correctness of the associated formula generator.

Model checkers are indeed relatively safe tools in the sense that when a model checkers takes a formula and answers "YES" one can be reasonably sure that there is a model for this formula. However it is much more difficult to be certain that the formula that has been given is a correct abstraction of the initial problem one wants to solve. The argument is usually a pen and paper proof in a scientific publication, that is: checked by human beings who can leave a few subtle errors.

If time permits one can try to apply the same methodology to a more intricate problem on autonomous robots with lights, in Euclidean spaces, and in particular on the validation, within Pactole, of a Model-Checking abstraction recently proposed by Defago et al\([5]\). This part may be done in collaboration with another internship.

Recommended previous knowledge:

- Formal proof, Proof assistants (Coq),
- Distributed Computing.
References


