Certified Impossibility Results for Byzantine-Tolerant Mobile Robots

Cédric Auger, Zohir Bouzid\textsuperscript{4}, Pierre Courtieu\textsuperscript{2}, Sébastien Tixeuil\textsuperscript{4,5}, and Xavier Urbain\textsuperscript{1,3}

1 École Nat. Sup. d’Informatique pour l’Industrie et l’Entreprise (ENSIIE), Évry, F-91025
2 CEDRIC – Conservatoire national des arts et métiers, Paris, F-75141
3 LRI, CNRS UMR 8623, Université Paris-Sud, Orsay, F-91405
4 UPMC Sorbonne Universités
5 Institut Universitaire de France

Abstract. We propose a framework to build formal developments for robot networks using the COQ proof assistant, to state and prove formally various properties. We focus in this paper on impossibility proofs, as it is natural to take advantage of the COQ higher order calculus to reason about algorithms as abstract objects. We present in particular formal proofs of two impossibility results for convergence of oblivious mobile robots if respectively more than one half and more than one third of the robots exhibit Byzantine failures, starting from the original theorems by Bouzid \textit{et al.}. Thanks to our formalisation, the corresponding COQ developments are quite compact. To our knowledge, these are the first certified (in the sense of formally proved) impossibility results for robot networks.

1 Introduction

Networks of static and/or mobile sensors (that is, robots) \cite{19} received increasing attention in the past few years from the Distributed Computing community. On the one hand, the use of cooperative swarms of inexpensive robots to achieve various complex tasks in potentially hazardous environments is a promising option to reduce human and material costs and assess the relevance of Distributed Computing in a practical setting. On the other hand, execution model differences warrant extreme care when revisiting “classical results” from Distributed Computing, as very small changes in assumed hypotheses may completely change the feasibility of a particular problem. Negative results such as impossibility results are fundamental in Distributed Computing to establish what can and cannot be computed in a given setting, or permitting to assess optimality results through lower bounds for given problems. Two notorious examples are the impossibility of reaching consensus in an asynchronous setting when a single process may fail by stopping unexpectedly \cite{18}, and the impossibility of reliably exchanging information when more than one third of the processes can exhibit arbitrary behaviour \cite{29}. As noted by Lamport, Shostak and Pease \cite{25}, correctly proving results in the context of Byzantine (a.k.a. arbitrary behaviour capable) processes is a major challenge, as they

\textsuperscript{*} This work was supported in part by the Digiteo Île-de-France project PACTOLE 2009-38HD. A preliminary version of this work appears as a 2-pages Brief Announcement in DISC 2013.
An attractive way to assess the validity of distributed algorithm is to use tool assisted verification, be it based on process algebra [4,20], local computations [27], Event-B [8], COQ [9], HOL [10], Isabelle/HOL [23], or TLA [25,24] that can enjoy an Isabelle back-end for its provers [13]. Surprisingly, only few works consider using mechanised assistance for networks of mobile entities, be it population protocols [14,11] or mobile robots [15,5]. In this paper, our goal is to propose a formal provable framework in order to prove positive or negative results for localised distributed protocols in mobile robotic networks, based on recent advances in mechanical proving and related areas, and in particular on proof assistants. Proof assistants are environments in which a user can express programs, state theorems and develop interactively proofs that will be mechanically checked (that is machine-checked). They have been successfully employed for various tasks such as the formalisation of programming language semantics [26,28], verification of cryptographic protocols [2], certification of RSA keys [31], mathematical developments as involved as the 4-colours [21] or Feit-Thompson [22] theorems.

**Our Contribution.** We developed a general framework relying on the COQ proof assistant to prove possibility and impossibility results about mobile robotic networks. The key property of our approach is that its underlying calculus is of higher order: instead of providing the code of the distributed protocols executed by the robots, we may quantify universally on those programs/algorithms, or just characterise them with an abstract property. This genercity makes this approach complementary to the use of model-checking methods for verifying distributed algorithms [7,11,15,17] that are highly automatic, but address mainly particular instances of algorithms. In particular, quantifying over algorithms allows us to express in a natural way impossibility results.

We illustrate how our framework allows for such certification by providing COQ proofs of two earlier impossibility and lower bound theorems by Bouzid et al. [6], guaranteeing soundness of the first one, and of the SSYNC fair version of the second one. More precisely, in the context of oblivious robots that are endowed with strong global multiplicity detection and whose movements are constrained along a rational line, and assuming that the demon (that is, the way robots are scheduled for execution) is fair, the convergence problem cannot be solved if respectively at least one half (Theorem 1) and at least one third (Theorem 2) of robots are Byzantine.

The interestingly short size of the COQ proofs we obtained using our framework not only makes it easily human-readable, but is also very encouraging for future applications and extensions of our framework.

**Related Work.** With reference to proof assistants, Küfner et al. [23] develop a methodology to develop ISABELLE-checked proofs of properties of fault-tolerant distributed algorithms in an asynchronous message passing style setting. This work’s motivations are similar to ours, however the setting (message passing distributed algorithms) is different, moreover it focuses on positive results only whereas we provide negative results, i.e. proofs of impossibility.

---

1 Distributed Robot model assumptions are presented in Section 2.