Brief Announcement: Automated Support for the Design and Validation of Fault Tolerant Parameterized Systems - A Case Study

Francesco Alberti[1], Silvio Ghilardi[2], Elena Pagani[2], Silvio Ranise[1], and Gian Paolo Rossi[2]

[1] FBK-Irst, Trento, Italia
[2] Università degli Studi di Milano, Milano, Italia

Background and motivations. Algorithms for ensuring fault tolerance are key ingredients in many applications such as avionics and networking. There is an increasing demand to integrate (formal) validation in the design process of these algorithms as they are often part of safety critical systems. When validation fails, the designer would benefit from tracking the sequence of events that led to an incorrect state to recover the error. To productively integrate formal verification in the design phase, tools should be able to return such error traces. Fault tolerant algorithms are often parametric, which makes their automated verification a daunting task. Indeed, checking that an algorithm satisfies a certain property requires to prove it *for any number of processes.*

Contributions. We propose the use of an infinite state model checker for safety properties, called MCMT [3] (`http://www.dsi.unimi.it/~ghilardi/mcmt`), to assist in the design of the considered class of algorithms. MCMT is particularly suitable for this purpose because it is based on a declarative framework in which parametric algorithms can be naturally specified by using first-order formulae I, Tr, and U for the set of initial states, the transitions, and the set of undesired states, respectively. The distinguishing feature of MCMT is that it applies the so-called *backward reachability procedure* [1] in a symbolic setting: a tree whose nodes are labeled by the formulae describing the pre-images of U with respect to the transitions in Tr is constructed and visited on-the-fly. The visit is interleaved with fix-point and safety checks so as to decide when the process can stop. To mechanize this, MCMT puts some *constraints on the format* of I, Tr, and U so that (a) the class of formulae describing the set of backward reachable states are closed under pre-image computation and (b) both fix-point and safety checks can be reduced to decidable logical problems, called Satisfiability Modulo Theories (SMT) [7] problems. Facts (a) and (b) rely on results precisely stated and proved in [2]; satisfiability tests involving *quantified formulae* are preprocessed by manual instantiations - subject to powerful but complete heuristics - and are then discharged via the integration with the SMT-solver YICES (`http://yices.csl.sri.com`).

The **first contribution** of our work is the definition of a sub-set of the formal framework [2] underlying MCMT, which is suitable for the specification of fault tolerant algorithms.

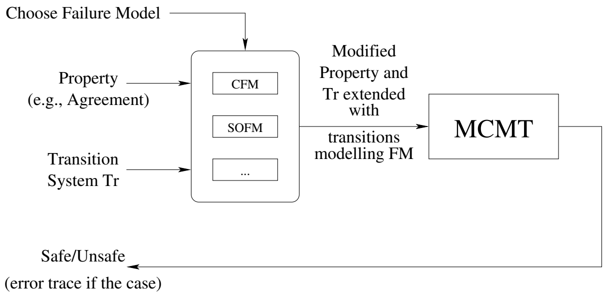
Choose Failure Model
Property (e.g., Agreement)
Transition System Tr
CFM
SOFM
...
Modified Property and Tr extended with transitions modelling FM
MCMT
Safe/Unsafe (error trace if the case)

Fig. 1. A methodology for designing fault tolerant algorithms

In order to describe the possible misbehaviors of real distributed systems, a taxonomy of failure models has been presented in the literature [6]: such taxonomy abstracts relevant features concerning faults that may occur in practice. For example, the simplest of such models is the Crash Failure Model (CFM) where processes may halt at any time. A slightly more realistic model consists of considering the possibility that a process may omit to send a message besides crashing at any time, called the Send Omission Failure Model (SOFM).

Since MCMT natively supports only the simplest failure model (i.e. the CFM) [5], the **second contribution** of our work is a technique to re-write the specification of an algorithm for a certain failure model, so as to represent more complex failures and enable the validation of distributed algorithms under critical conditions. The underlying idea is to add a local state variable to each process as a flag signaling whether the process is faulty or not, and to enlarge the specification with faulty behaviors. Then, we propose a design methodology for parametrized and fault tolerant algorithms (see Figure 1) that exploits the previous two contributions consisting of (i) specifying the algorithm and its safety property, (ii) choosing a failure model (e.g., the CFM or the SOFM), (iii) invoking MCMT, and (iv) modifying the specification of the algorithm according to the analysis of the error trace returned by the tool (if any), before repeating the procedure. This schema can be easily blended with a standard incremental and iterative approach to design.

The **third contribution** is to apply this methodology to replay the design of the reliable broadcast algorithms of Chandra and Toueg in [8]. We focus on Agreement as the safety property to check. The authors of [8] consider several parametric algorithms, obtained by stepwise refinement. Table 1 shows the results of our experiments. Algorithm 1 is the simplest one, designed to be correct with the CFM (first column of Table 1). Its unsafety w.r.t. the SOFM is quickly established by MCMT by finding a shorter error trace than the one described in [8] (second column of Table 1). Algorithm 1e is a first refinement of Algorithm 1, which is still found to be unsafe. In this case, the error trace found by MCMT, after a manual analysis, corresponds to that described in [8]. Algorithm 2 is the second refinement and its up-front verification turned out to be quite problematic, even using MCMT invariant synthesis capabilities [4]. Fortunately,

Table 1. MCMT performances

	Algo. 1, CFM	Algo. 1, SOFM	Algo. 1e, SOFM	Algo. 2, SOFM
Safe (agreement)	Yes	No	No	Yes
time (sec)	1.18	17.66	1,709.93	4,719.51
# state vars	8	9	11	15
# transitions	13	13+3	16+6	22+6
# nodes	113	464	9,679	11,158
# SMT calls	2,792	20,009	1,338,058	2,558,986
Length unsafe trace	×	11	33	×
# invariants	×	×	×	19 (+**7**)

Timings obtained on an Intel Core Duo 2.66 GHz with 2 GB, running Debian Linux. (The complete specifications of the algorithms considered here can be downloaded at `http://www.falberti.it/reliableBroadcast`.)

the possibility to interact with MCMT allowed us to add 7 more system properties (e.g., "there is only one coordinator at a time") to the specification. These properties, and 19 more invariants automatically found by the tool, have been validated by MCMT before their use, thus guaranteeing that their inclusion does not affect the main verification result. On the other hand, their adoption significantly pruned the backward search tree and yielded the performances reported in the last column of Table 1. The tool also validated the three lemmata used in [8] to perform the pen-and-paper proof of the correctness of the algorithm. These results show the practical viability of our technique.

There are two main lines for future work. First, we would like to consider more general failure models (e.g., general omission) to conclude the formal validation of the reliable broadcast algorithms in [8]. Second, we intend to refine our models so as to consider temporal constraints that would widen the scope of applicability of our techniques to more realistic algorithms.

References

1. Abdulla, P.A., Cerans, K., Jonsson, B., Tsay, Y.-K.: General decidability theorems for infinite-state systems. In: Proc. of LICS, pp. 313–321 (1996)
2. Ghilardi, S., Nicolini, E., Ranise, S., Zucchelli, D.: Towards SMT Model-Checking of Array-based Systems. In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS (LNAI), vol. 5195, pp. 67–82. Springer, Heidelberg (2008)
3. Ghilardi, S., Ranise, S.: Mcmt: A Model Checker Modulo Theories. To appear in Proc. of IJCAR (July 2010)
4. Ghilardi, S., Ranise, S.: Goal Directed Invariant Synthesis for Model Checking Modulo Theories. In: Giese, M., Waaler, A. (eds.) TABLEAUX 2009. LNCS (LNAI), vol. 5607, pp. 173–188. Springer, Heidelberg (2009)
5. Ghilardi, S., Ranise, S.: A note on the stopping failure model (Unpublished note), `http://homes.dsi.unimi.it/~ghilardi/mcmt/stop_fail_note.pdf`
6. Lynch, N.A.: Distributed Algorithms. Morgan Kaufmann, San Francisco (1996)
7. The SMT-LIB initiative, `http://www.SMT-LIB.org`
8. Toueg, S., Chandra, T.D.: Time and Message Efficient Reliable Broadcast. In: van Leeuwen, J., Santoro, N. (eds.) WDAG 1990. LNCS, vol. 486, pp. 289–303. Springer, Heidelberg (1991)