Learning to prove safety over parameterised concurrent systems

Yu-Fang Chen[1], Chih-Duo Hong[2], Anthony W. Lin[2], and Philipp Rümmer[3]

[1] Academia Sinica
[2] Oxford University
[3] Uppsala University

Parameterised concurrent systems are infinite families of finite-state concurrent systems, parameterised by the number n of processes. Examples of parameterised concurrent systems include models of distributed algorithms which are typically designed to handle an arbitrary number n of processes. Verification of such systems then amounts to proving that a desired property holds for *all* permitted values of n. For example, proving that the safety property holds for a dining philosopher protocol entails proving that the protocol with any given number n of philosophers ($n \geq 3$) can never reach a state when two neighbouring philosophers eat simultaneously. For each given value of n, verifying safety is decidable, albeit the exponential state-space explosion in the parameter n. However, when the property has to hold for each value of n, the number of system configurations a verification algorithm has to explore is potentially infinite.

Regular model checking is a well-known generic framework for modelling parameterised concurrent systems. One standard technique in regular model checking to prove safety is by exhibiting an *inductive invariant*, i.e., a set *Inv* of configurations satisfying (i) *Inv* is closed under application of the transition relation, (ii) *Inv* subsumes the set *Init* of all initial configurations, and (iii) *Inv* does not intersect with the set *Bad* of unsafe configurations. In regular model checking, the sets *Init* and *Bad* are given as regular languages and the transition relation is represented by a regular transducer. For this reason, it is decidable to check a candidate regular set against conditions (i)-(iii). A natural method to prove safety in regular model checking is thus to exhibit the existence of a *regular* inductive invariant.

In this talk, we shall present a simple and practical solution to synthesise regular inductive invariants in regular model checking. Our solution exploits Angluin's classic L^* learning algorithm and its variants, which learn a regular language by making membership and equivalence queries to a user-provided teacher. To answer membership queries, we propose to restrict to *length-preserving* transition relations. In theory, length-preservation is not a restriction for safety analysis, since it just implies that each instance of the considered parameterised system is operating on bounded memory of size n (but the parameter n is unbounded). Experience shows that many practical examples in parameterised concurrent systems can be captured naturally in terms of length-preserving systems. The benefit of the restriction is that the problem of membership queries is now decidable, since the set of configurations reachable from any given configuration is finite and can be solved by a standard finite-state model checker. To answer

equivalence queries, we propose that a *strict but generous* teacher be employed in L^* learning for regular inductive invariants. The teacher is strict in the sense that he attempts to teach the learner the minimal inductive invariant (be it regular or not), and is generous in the sense that he is satisfied when the candidate answer posed by the learner is an inductive invariant without being minimal. For this reason, when the learner asks whether w is in the desired inductive invariant, the teacher will reply NO if w is not reachable from *Init*. The same goes with an implication counterexample (v, w), for which the teacher will say that an unreachable v is not in the desired inductive invariant. Our solution is guaranteed to terminate when the set of reachable configurations is regular.

We have tested our solution on standard as well as new examples in regular model checking, including the dining philosopher protocol, the dining cryptographer protocol, and several mutual exclusion protocols (e.g. Bakery, Burns, Szymanski, and German). Our experiments show that, despite the simplicity of our solution, it can perform at least as well as many sophisticated safety analysis algorithms developed in the past fifteen years. The full paper of this work can be found at [1].

References

1. Yu-Fang Chen, Chih-Duo Hong, Anthony Widjaja Lin, and Philipp Rümmer. Learning to prove safety over parameterised concurrent systems. In submission.