Author: Mariëlle Stoelinga
Title: *Alea jacta est:* verification of probabilistic, real–time and parametric systems
Language: English, summary in Dutch
Supervisor: Frits Vaandrager
Date of Defense: April 22, 2002
Institution: University of Nijmegen, the Netherlands

Abstract

Motivation During the past decades, enormous progress has been made in concurrency theory. Models and techniques for the specification and analysis of the basic, functional correctness properties are well-understood and have found their way into industry. Hence, researchers have taken up the challenge to investigate more complex types of systems, for instance systems whose behavior is continuous (hybrid systems), randomized (stochastic/probabilistic systems), mobile and dynamically reconfigurable (higher order systems), etc.

In this thesis, we study concurrent systems with *probabilistic*, *parametric* and *timing* aspects, with the emphasis on probabilistic systems. More specifically, we restrict ourselves to discrete probabilities, timing parameters and the real-time base. These features allow, in particular, a more accurate analysis of distributed algorithms and communication protocols.

Main Results The main results of this thesis can be summarized as follows.

- We present a testing scenario that justifies the notion of external behavior for (closed) probabilistic systems that has been proposed in the literature.
- We develop a type of probabilistic (bi-)simulation that abstracts from internal computation in some sense and that can be computed efficiently.
- We present a model checking technique capable of synthesizing parameter constraints for the correctness of timed systems.
- We prove the correctness of the IEEE 1394 Root Contention Protocol using techniques developed in this thesis. We analyze the probabilistic, timing and parametric behavior of this protocol.

Overview We state our results in the probabilistic automaton (PA) framework, developed by Segala. PAs are just labeled transition systems (also called state machines or automata), where a transition no longer leads to a single state, but chooses a next state with a certain probability. The behavior of a PA is given by the set of its *trace distributions*, which are the equivalents of the traces (infinite words over the action alphabet) in a labeled transition system. The implementation relation proposed for closed PAs is *trace distribution inclusion*. This means that we consider one PA, expressing the implementation of system,

to be a correct implementation of another PA, expressing the specification, if every trace distribution of the first is also a trace distribution of the second.

In this thesis, we justify the trace distribution inclusion as an implementation relation via an intuitive testing scenario. That is, we present an alternative characterization of this relation in terms of events perceived by an external observer. We show how the trace distributions of a PA can be observed by analyzing a number of independent runs. By applying statistic methods to the frequencies of these runs, one can distinguish between sets of runs produced by the PA and sets of runs not produced by it. Since, in the end, the behavior of a system is what one can observe, the testing scenario shows that the trace distributions are indeed a plausible notion of external behavior.

For practical verification purposes, however, the trace distribution inclusion relation is not very useful, because it takes into account the global behaviors of two systems, in terms of executions. Simulation relations, instead, compare the local behavior of systems in terms of their states and transitions. As a result, simulation relations are sound for establishing trace distribution inclusion and, importantly, are much easier to verify.

This thesis introduces a new type of (bi-)simulation relations for PAs, called *delay (bi-)simulation*. We consider four variants of these and present alternative characterizations in terms of norm functions. All delay relations abstract, although in a limited way, from the internal computations of a system. This abstraction is desirable if these computations are irrelevant for the verification question one wants to tackle. On the other hand, the limited form of abstraction allows the delay relations to be computed by efficient algorithms.

Most concurrent systems are parametrized in some way: parameters are for instance number of processes in the application, the communication delay, the clock speed, the network topology, etc. Usually, these systems work correctly for certain values of the parameters, whereas for others they fail. Therefore, it is relevant to derive the exact parameter constraints which are needed for correct system operation.

We present a model checker that is able to synthesize parameters constraints for linear parametric timed automata (PTAs). Model checkers have proven to be very successful in the automatic verification of concurrent systems and in this thesis we extend one of the classical model checking algorithms (that is, the forward reachability algorithm) for timed automata to PTAs. The algorithm has been implemented on top of the existing model checker Uppaal and the performance of our prototype implementation compares favorably with similar tools. Furthermore, we identify the class of lower bound/upper bound automata, for which parametric model checking is much easier.

Probabilities, timing and parameters come together in the verification of the IEEE1394 Root Contention protocol. This is an industrial leader election protocol for two processes. We analyze two different models. First, we present a protocol model that includes real-time, probabilities and parameters. The correctness of this model is proven by hand, using simulation relations introduced earlier (namely, step and hyperstep refinements). We derive two parameter constraints that guarantee correct protocol operation. Second, we present an enhanced model, which describes the inter-process communication more realistically. For this model, we investigate the real-time and parametric properties. We derive the constraints "experimentally," by using the model checker Uppaal to check a large number of protocol instances for fixed parameter values. Due to the differences in the communication models, the constraints inferred for the enhanced model are stronger than the ones found for the first model. Finally, we use the parametric extension of Uppaal to carry out rigorous parameter analysis for the enhanced model. The findings confirm the results obtained experimentally.

Table of Contents

Author's correspondence address

Mariëlle Stoelinga
Department of Computer Engineering
University of California
Santa Cruz, CA 95064
USA

`marielle@soe.ucsc.edu`
`http://www.soe.ucsc.edu/~marielle`