Chapter 7

Online Testing of Nondeterministic Systems with the Reactive Planning Tester

Jüri Vain
Tallinn University of Technology, Estonia

Marko Kääramees
Tallinn University of Technology, Estonia

Maili Markvardt
Tallinn University of Technology, Estonia

ABSTRACT

We describe a model-based construction of an online tester for black-box testing. Contemporary model-based online test generators focusing mainly on computationally cheap but far from optimal planning strategies cover just a fraction of the wide spectrum of test control strategies. Typical examples of those used are simple random choice and anti-ant. Exhaustive planning during online testing of nondeterministic systems looks out of reach because of the low scalability of the methods in regard to the model size. The reactive planning tester (RPT) studied in this chapter is targeted to fill the gap between these two extremes. The key idea of RPT lies in offline static analysis of the IUT (implementation under test) model to prepare the data and constraints for efficient online reactive planning. The external behavior of the IUT is modelled as an output observable nondeterministic EFSM (extended finite state machine) with the assumption that all the transition paths are feasible. A test purpose is attributed to the transitions of the IUT model by a set of Boolean variables called traps that are used to measure the progress of the test run. We present a way to construct a tester that at runtime selects a suboptimal test path from trap to trap by finding the shortest path that covers unvisited traps within planning horizon. The principles of reactive planning are implemented in the form of the decision rules of selecting the shortest path.
paths at runtime. Based on an industrial scale case study, namely the city lighting system controller, we demonstrate the practical use of the RPT for systems with high degree of nondeterminism, deep nested control loops, and requiring strictly bounded tester response time. Tuning the planning horizon of the RPT allows a trade-off to be found between close to optimal test length and scalability of tester behavior with computationally feasible expenses.

INTRODUCTION

Model-Based Testing is the automatic generation of efficient test procedures/vectors using models of system requirements and specified functionality. Specific activities of the practice are (1) Build the model, (2) Generate expected inputs (3) Generate expected outputs, (4) Run tests, (5) Compare actual outputs with expected outputs, and (6) Decide on further actions (whether to modify the model, generate more tests, or stop testing, estimate reliability (quality) of the software (DACS Gold Practice Website, 2010).

On-Line Testing

On-line testing is widely considered to be the most appropriate technique for model-based testing (MBT) of embedded systems where the implementation under test (IUT) is modelled using nondeterministic models (Veane, Campbell, & Schulte, 2007; Veane, Campbell, Grieskamp, Schulte, Tillmann, & Nachmanson, 2008). Nondeterminism of IUT models stems from the physical nature of the IUT, particularly, its internal parallel processes, timing conditions, and hardware-related asynchrony of executing the processes. Other sources of model nondeterminism are the higher abstraction level of the model compared to IUT implementation and the ambiguities in the specifications of the IUT. Often, the term \textit{on-the-fly} is used in the context of on-line testing to describe the test generation and execution algorithms that compute and send successive stimuli to IUT incrementally at runtime. Computation of test stimuli is directed by the test purpose and the observed outputs of the IUT.

The state-space explosion problem experienced by many model-based offline test generation methods is avoided by the on-line techniques because only a limited part of the state-space needs to be kept track of at any point in time when a test is running. However, exhaustive planning would be difficult on-the-fly because of the limitations of available computational resources at the time of test execution. Thus, developing a planning strategy for industrial strength online testing should address in the first place the trade-off between reaction time and on-line planning depth to reach the practically feasible test cases.

The simplest approach to on-the-fly selection of test stimuli in model-based on-line testing is to apply so called random walk strategy where no computation sequence of IUT has an advantage over the others. The test is performed usually to discover violations of input/output conformance relation IOCO (Tretmans, 1999) or timed input/output conformance relation TIOCO (Brinksma & Tretmans, 2001) between the IUT and its model. Random exploration of the state space may lead to test cases that are unreasonably long and nevertheless may leave the test purpose un-achieved. On the other hand, the long test cases are not completely useless, some unexpected and intricate bugs that do not fit under well-defined test coverage criteria can be detected when a test runs hours or even days.

In order to overcome the deficiencies of long lasting testing usually additional heuristics, e.g. “anti-ant” (Li & Lam, 2005; Veane, Roy, & Cambell, 2006), dynamic approach of DART system (Godefroid, Halleux, Nori, Rajamani, Schulte, Tillmann, & Levin, 2008), inserted assertions (Korel & Al-Yami, 1996), path fitness