Abstract. Büchi automata have been widely used for specifying linear temporal properties of reactive systems and they are also instrumental for designing efficient model-checking algorithms. In this paper we extend specification-based testing to Büchi automata. A key question in specification-based testing is how to measure the quality (relevancy) of test cases with respect to system specification. We propose two state coverage metrics for measuring how well a test suite covers a Büchi-automaton-based requirement. We also develop test generation algorithms that use counter-example generation capability of an off-the-shelf model checker to generate test cases for the coverage criteria inferred by these metrics. In our experiment we demonstrate the feasibility and performance of the coverage criteria and test generation algorithms for these criteria. In [13] we proposed testing coverage metrics and criteria for properties in Linear Temporal Logic (LTL) and referred to the new approach as property-coverage testing. This research shares the same motivation as in [13] and it extends property-coverage testing to specifications in Büchi automata. Since automaton minimization techniques can be used to reduce the structural diversity of semantically equivalent Büchi automata, we argue that a coverage metric based on Büchi automata is less susceptible to syntactic changes of a property than a LTL-based coverage metric, and hence the proposed coverage metrics measure the relevancy of a test suite to the semantics of a linear temporal property. We also discuss an algorithm for refining a Büchi-automaton-based requirement based on its strong state coverage metric. Our experiment demonstrates the feasibility and performance of our coverage criteria and test generation algorithms.

1 Introduction

Testing and formal verification are considered as two important and yet complementary methods in verifying and validating reactive systems. Reactive systems refer to those dynamic systems that continuously operate and interact with their environment. Examples of reactive systems include engine control modules (ECMs) in automobiles, and autopilot modules in airplanes. Ensuring the correct functioning of reactive systems is of uttermost interest to automobile, aerospace, and many other industries where reactive systems are widely deployed in safe- and/or
mission-critical applications. Whereas testing is to check the behavior of a system under a controlled environment, formal verification is to algorithmically analyze a system. A critic of testing may be best summarized by Dijkstra’s notable statement “testing shows the presence, not the absence of bugs”. Nevertheless, despite this shortcoming, testing can work where formal verification stops short. Compared with formal verification, testing usually has a better scalability, and it can be applied to implementation directly, for instance, in a setting such as hardware-in-loop testing. In the foreseeable future testing will continue to play a predominant role on validating and verifying reactive systems.

A direction of our research is to study how to harness the synergy of testing and formal verification. For instance, formal verification techniques such as model checking (cf. [3]) have enjoyed a great deal of successes in past two decades. As the result, rigorous formalisms such as temporal logics and Büchi automata are increasingly popular for specifying requirements for high-dependable reactive systems. A research question is how testing can benefit from the proliferation of these high-quality formal specifications. One of important formal verification techniques is linear temporal model checking (cf. [5]), in which a system design is checked against a linear temporal property. Büchi automata have been used for specifying linear temporal properties. Since other formalisms such as Linear Temporal Logic (LTL) can be translated into Büchi automata, Büchi automata are also used as a unified theoretical platform for reasoning about linear temporal model checking algorithms. For instance, efficient linear temporal model checking algorithms such as those in [5,6] have been proposed with the use of Büchi automata.

The purpose of this research is to develop a framework for specification-based testing with Büchi automata. We will address the following issues:

1. We need to define the relevancy of a test case to a specification in a Büchi automata. For this purpose, we propose two variants of state coverage metrics that measure how a Büchi automaton is covered during a test. A weak variant indicates that a particular state may be reached during a test, whereas a strong variant requires that the state must be reached during a test;
2. We need to develop a practical way to produce test cases for the testing criteria inferred by the proposed metrics. For this purpose, we propose the algorithms that can use the counterexample mechanism of an off-the-shelf model checker to generate test cases for state coverage criteria.

In addition, we consider property refinement based on the proposed metrics. Lack of state coverage may be caused by incorrect/incomplete implementation, and/or imprecise/loose specification. Whereas testing helps identify the first problem, a careful examination and refinement of the specification will address the latter issue. We will discuss how to enhance our test generation algorithms to systematically refine a property expressed as a Büchi automaton.

The rest of the paper is organized as follows: Section 2 prepares the notations that will be used in the rest of the paper; Section 3 proposes two variants of state coverage metrics for Büchi automata; Section 4 gives the algorithms for generating tests for state coverage using a model checker; Section 5 discusses