

Checking Experiment Design Methods by Michal Soucha

A system modelled by a *Finite-State Machine* (FSM) is verified against its *specification* using a suite of input sequences. *Testing methods* create such a *test suite* based on the specification automatically. The main challenge is to design a testing method producing a test suite as short as possible due to time and cost of an application of tests on the system *implementation*. We propose a new state-of-the-art testing method that creates a test suite of a single sequence which is normally called a *Checking Sequence*.

The process of verification, or a *checking experiment*, is shown in the example below. Testing methods should create a test suite that detects any errors that may occur in the implementation. However, if it does not, our proposed *Fault Coverage Checker* can find all errors that the proposed suite is not able to reveal. The thesis also includes an enhancement of concept of Finite-State Machines and current Testing methods.

Importance of research

Checking experiment has a wide variety of use:

- software verification
- hardware components
- computer security
- network protocols
- business processes
- *Automated Active Learning*

The proposed M-method

- new state-of-the-art method
- sketch of a formal proof
- experimentally proven

Additional Contribution

- new parallel approaches for Separating Sequences design
 - reduction of Characterizing Set in the number of Separating Sequences
- Testing Methods employ Characterizing Set to design a checking experiment

Example: A turnstile



Actions:

Coin (C), Push (P)

Responses:

Free (F), Locked (L), No output (N)

Testing Method

We use our new method in the example

designs
a test
suite

Checking Sequence
(a series of actions)

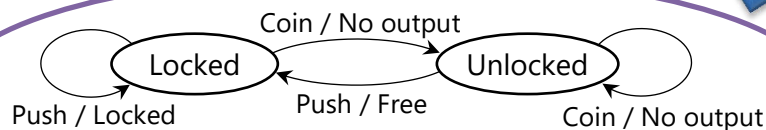
PPCPCPP

LLNFNNFL

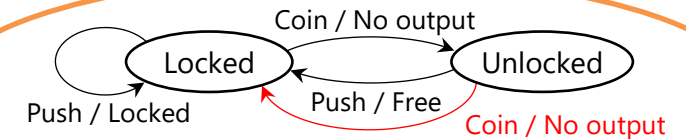
LLNFNNLL

Error Detection

Specification



Implementation



Fault Coverage Checker

- improved checker of testing methods
- passive learning approach
- reconstructs the specification from a given test suite or provides all faulty machines that the suite does not cover

inputs for