Bachelor Thesis Topic

Probabilistic Typestates Analysis by Testing

Motivation and Background

Typestate analysis is a form of program analysis involved with understanding valid sequences of operations that a software element may carry out. Because of this, it has also been called protocol analysis, as it serves well to document protocol action sequences.

Typestates have been used successfully as a means to understanding the behaviour of an undocumented piece of software. Some approaches have built these typestates automatically via static analysis, resulting in overapproximations (i.e., the typestate model allows more sequences than the actual ones).

A way to obtain underapproximations is via testing, which can only evidence what has been tested. A combination of both may allow a tighter approximation to the real typestate (obtaining the exact typestate is in general undecidable). This Bachelor thesis involves extending a current typestate-by-testing approach into another one that includes statistical analysis to probabilistically quantify the model, thus giving a deeper understanding of the protocol.

Goals

The goal of the research is to extend a typestate-by-testing tool in order to obtain a probabilistic model that represents the frequency with which the protocol actions are invoked.

Description of the Task

The tasks of the thesis are

- to extend and implement a probabilistic approach to typestates
- to understand the impact of parameter selection in the building of the typestate

Research Type

Theoretical Aspects: *****

Industrial Relevance: *****

Implementation *****

Prerequisite The student should be enrolled in the master of software engineering or master of computer science program, and has completed the required course modules to start a master thesis.

Skills required The student should have or be willing to acquire the following skills: graph theory, statistics. The student should also have, or want to learn, programming in Java and Clojure.

References

[1] Strom, R. Yemini, S. Typestate: A programming language concept for enhancing software reliability. IEEE Trans. Softw. Engin, 1986

[2] de Caso, Garbervetsky et. al., Enabledness-based program abstractions for behavior validation, ACM TOSEM 2013

[3] Bonfiglio, Nussembaum, MENTAT Una herramienta de validación dinámica para código Java, MSc. Thesis, UBA https://github.com/bauna/Mentat

Contacts

Esteban Pavese, Humboldt-Universität zu Berlin, Institut für Informatik, Lehrstuhl Software Engineering, Unter den Linden 6, 10099 Berlin, Germany

Application Please contact me during my office hours or write an email with the title: “[Thesis]-Probabilistic Slicing” to se-career@informatik.hu-berlin.de