Model checking approach to the correctness proof of complex systems

Marina M. Alekseeva, Ekaterina A. Dashkova

First-year Masters of Information Systems Chair
Yaroslavl State University

28^{th} April 2011

Outline

Introduction

- Correctness of Information and Communication Technology (ICT) systems is the background for their safety.
- The key instrument for design process is verification techniques.
- Model checking is one of various verification techniques.

Outline

Model-checking

- The accurate modeling of systems often leads to the discovery of incompleteness, ambiguities, and inconsistencies in informal system specifications.
- The system model is usually automatically generated from a model description that is specified in some appropriate dialect of programming or hardware description languages.
- Models are mostly expressed using finite-state automaton, consisting of a finite set of states and a set of transitions.
- Simulation can be used effectively to get rid of the simpler category of modeling errors.

Outline

Types of programming systems

Transforming systems

Finite automaton in programming traditionally used in design of compilers. In this situation automaton is understood as some calculating feature which has an input line and output line.

Reactive systems

In this case automaton is a device that has several parallel input lines (often binary), on which in real time the signals from the environment is coming. Processing such kind of signals, automaton is forming values for several parallel outputs.

"Complex behavior"

- compilers,
- archivators,
- telecommunication systems,
- systems of control and managing of physical devices.
- Transition systems (TS) are fundamental instrument for modeling software and hardware systems.

Example

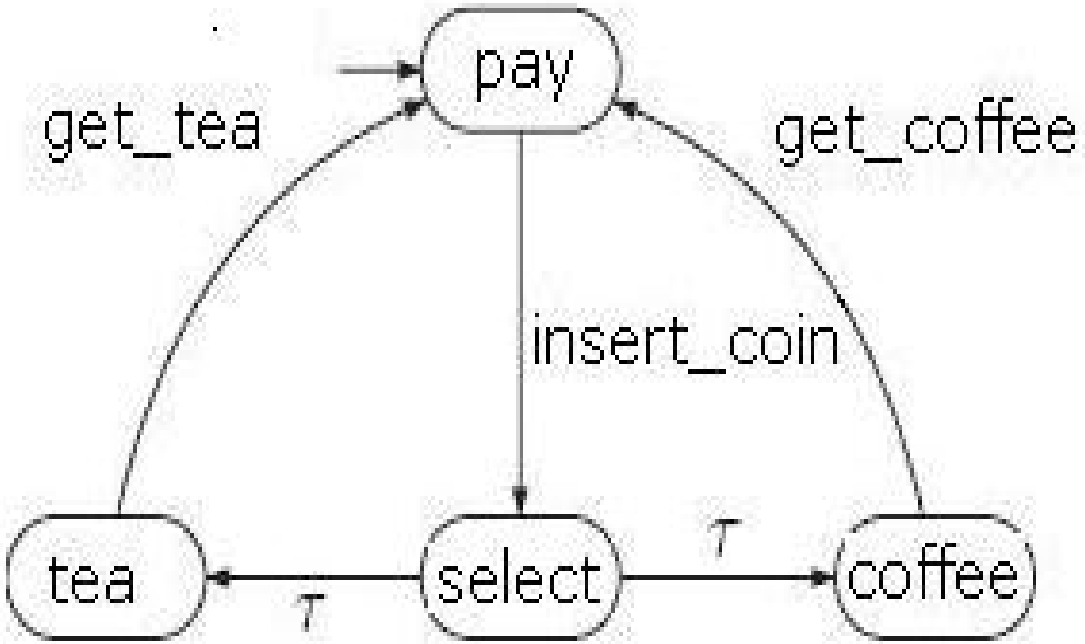

Figure: A simple transition system

Example

- The state space is $S = \{pay, select, tea, coffee\}$.
- Act is a set of actions:
  $Act = \{insert\_coin, get\_tea, get\_coffee, \tau\}$.
- AP is a set of atomic propositions.
- TS is called finite if S, Act, and AP are finite.

Outline

Conclusion

- Theory of programming even in the 1968 openly accepted the crisis of software development.
- Theoreticians and practitioners of software underline that the crisis of methods of the development of software shows mainly during the design of the systems with complex behavior.
- Automata-based approach can deal with this problem.

The bibliography

- Anikeev M., Madlener F., Schlosser A., Huss S.A., Walter C., ”Automated Correctness Proof of Algorithm Variants in Elliptic Curve Cryptography”
- Baier Christel, Katoen Joost-Pieter. ”Principles of Model Checking”
- Egor V. Kuzmin, ”Introduction to the theory of mathematical processes and structures”
- N.I. Polikarpova, A.A. Shalyto, ”Automata-based programming”