A Model Projection Technique
for Compositional Verification using Model Checking

Dong-Ah Lee and Junbeom Yoo
KONKUK University
Contents

• Introduction

• The Model Projection Technique

• Case Study

• Conclusion
Introduction

• Model checking

• Checking whether a model meets a specification exhaustively and automatically
• Demonstrating function correctness of hardware or software systems
Current Approaches & Limitations

- Code slicing (program slicing)
  - Program decomposition by analyzing their features
  - Slicing by static analysis
    - Data flow, control flow, etc.
  - Independent checking of the sliced programs, but no overall views
  - Code slicing by static analysis does not care about dynamic behavior

- Abstraction
  - Making an abstract model of a program
    - Abstraction on the variables, Abstraction by restriction, etc.
  - The bigger abstraction, the bigger gaps
  - Abstraction of large-scale software for model checking makes the big gap
A technical process to identify relevant parts of source code with a verification purpose by executing a model of software

Identification of relevant parts by dynamic analysis

More concrete models derived from source code

1. Formal modeling at system-level \((FM_S)\)
2. Generating simulation scenarios \((S_i)\)
   - Selecting the model checkers of \(fm_i\)
3. Simulating system-level model
4. Identifying running parts \((P_i)\)
5. Tracing the parts to source code \((C_i)\)
6. Model checking of the parts \((fm_i)\)
Case Study

- Qplus-AIR
  - a RTOS for avionics complying the ARINC 653 by ETRI

ARINC 653 Specification
Qplus-AIR SRD
Qplus-AIR SDD

Qplus-AIR Architecture
Case Study

• Formal modeling at system-level using Statemate
Case Study

• Simulation Scenarios
  • Change partition modes

2.3.1.4 Partition Modes

The SET_PARTITION_MODE service allows the partition to request a change to its operating mode. The Health Monitor, through its health monitoring configuration data, can also instigate mode changes. The current mode of the partition is available with the GET_PARTITION_STATUS service.

Partition modes and their state transitions are shown in the following diagram:

![Diagram showing state transitions between Cold_Start, Warm_Start, Idle, and Normal modes]
Case Study

- Simulation of the System-level Model with the Scenarios
Case Study

- Traceability Analysis from Model to Source Code
Case Study

- CBMC
  - Bounded model checking for ANSI-C programs
  - Checking pointer safety, array bound, overflow, divided by zero, etc.
  - User defined assertion checking
Case Study

- Property specification
  - All the defined and *undefined* mode changes
Case Study

- Found violations among undefined mode changes

<table>
<thead>
<tr>
<th>Verification Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Predecessor Mode</strong></td>
<td><strong>Successor Mode</strong></td>
</tr>
<tr>
<td>COLD_START</td>
<td>WARM_START</td>
</tr>
<tr>
<td>IDLE</td>
<td>IDLE</td>
</tr>
<tr>
<td>IDLE</td>
<td>NORMAL</td>
</tr>
<tr>
<td>NORMAL</td>
<td>NORMAL</td>
</tr>
</tbody>
</table>

**VIOLATION: IDLE → IDLE**
Other Case Studies

• Verification of communications
  • Model checker SPIN: the formal verification of multi-threaded SW applications

• Verification of a scheduler in Qplus-AIR
  • TIMES: Modeling, Verification, and Implementation of Embedded Systems
Conclusions

• The model projection technique to identify relevant parts with verification purposes
• Compositional verification with systematic analysis about relation and influence between components at a system level

FUTURE PLAN

• An elaborate model and a tool for traceability analysis to make projection much easier and quicker