CS453: Automated Software Testing

Moonzoo Kim
Software Testing and Verification Group
CS Dept. KAIST
Strong IT Industry in South Korea

Time-to-Market?

SW Quality?

Moonzoo Kim
### Embedded Software in Two Different Domains

<table>
<thead>
<tr>
<th></th>
<th>Consumer Electronics</th>
<th>Safety Critical Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Examples</strong></td>
<td>Smartphones, flash memory platforms</td>
<td>Nuclear reactors, avionics, cars</td>
</tr>
<tr>
<td><strong>Market competition</strong></td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Life cycle</strong></td>
<td>Short</td>
<td>Long</td>
</tr>
<tr>
<td><strong>Development time</strong></td>
<td>Short</td>
<td>Long</td>
</tr>
<tr>
<td><strong>Model-based development</strong></td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Important value</strong></td>
<td>Time-to-market</td>
<td>Safety</td>
</tr>
</tbody>
</table>

- Conventional Testing
- Concolic testing
- Model checking

---

Moonzoo Kim
Role of S/W: Increased in Everywhere

Percent of functionality provided by software

Year of introduction:
- 1960년: F-4
- 1970년
- 1982년
- 2000년: F-22

자료출처: Watts Humphrey 2002
Motivation: Poor Quality of SW
Current Practice for SW

- SW developers have to follow **systematic disciplines** for building and analyzing software with high quality
  - This class focuses on the analysis activities
Software Development Cycle

- A practical end-to-end formal framework for software development
Main Target Systems

- Embedded systems where highly reliable SW technology is a key to the success
  - The portion of SW in commercial embedded devices increases continuously
  - More than 50% of development time is spent on SW testing and debugging
How to Improve the Quality of SW

1. Systematic testing (can be still manual)
   - Coverage criteria
   - Mutation analysis

2. Testing through automated analysis tools
   - Scientific treatment of SW with computing power
   - Useful tools are available

3. Formal verification
   - Guarantee the absence of bugs
Questions???

• Is automated testing really beneficial in industry?
  – Yes, dozens of success stories at Samsung

• Is automated testing academically significant?
  – Yes, 3 Turing awardees in ‘07

• Is automated testing too hard to learn and use?
  – No, there are tools available
Companies Working on Software Verification
Verification of High-Availability Protocol

- We develop a formal model of high-availability protocol used in commercial security appliances
  - HA protocol coordinates a group of firewalls
- We found several problems in HA regarding a master election procedure
Home Service Robot

- Designed for providing various services to human user
  - Service areas: home security, patient caring, cleaning, etc.
OneNAND® Flash Memory

- Each memory cell can be written limited number of times only

- XIP by emulating NOR interface through demand-paging scheme

- Performance enhancement
We have developed CONcrete and symBOLic (CONBOL) framework that is an automated concolic unit testing tool based-on CREST-BV for embedded software.
Research Trends toward Quality Systems

• Academic research on developing embedded systems has reached stable stage
  – just adding a new function to a target system is not considered as an academic contribution anymore

• Research focus has moved on to the quality of the systems from the mere functionalities of the systems
  – Energy efficient design, ez-maintenance, dynamic configuration, etc

• Software reliability is one of the highly pursued qualities
  – ASPLOS 2011 Best paper
    • “S2E: a platform for in-vivo multi-path analysis for software systems” @ EPFL
  – OSDI 2008 Best paper
    • “Klee: Unassisted and Automatic Generation of High-Coverage Tests for Complex Systems Programs” @ Stanford
  – NSDI 2007 Best paper
Tool-based Interactive Learning

- **Model checker**
  - Explicit model checker: Spin home page
  - Symbolic model checker: NuSMV home page
- **Software model checker**
  - Bounded model checker for C program: CBMC home page
  - Predicate abstraction for C program: BLAST home page
- **Satisfiability solver**
  - MiniSAT home page
- **Satisfiability Module Solver**
  - Yices home page
  - Z3 home page
- **Concolic testing tools**
  - CREST home page
Class Schedule

- wk1: overview on automated SW analysis techniques
- Wk2-3: coverage based SW
- wk4: background on Propositional logic and SAT (Satisfiability) solvers
- wk5: SAT solver heuristic and tool application 1: MiniSAT
- wk6: background on First order logic
- wk7: Satisfiability Modulo Theory (SMT) basic
- wk8: midterm exam

- wk9: advanced application of SMT solvers
- wk10: directed automated random testing
- wk11: tool application: CREST
- wk12: basic temporal logic for requirement property
- Wk13-14: tool application: Spin & NuSMV
- wk15: state space minimization techniques
- wk16: final exam
Final Remarks

• For undergraduate students:
  – Highly recommend URP studies or independent studies
    • Ex. 이준희 (05학번) got a silver award and macbook air notebook 😊
      – Debugging Linux kernel through model checking to detect concurrency bugs
    • Ex2. Nam Dang wrote down a paper on distributed concolic testing
      – Y.Kim, M.Kim, N.Dang, Scalable Distributed Concolic Testing: a Case Study on a Flash Storage Platform, Verified Software Track @ Intl. Conf. on Theoretical Aspects of Computing (ICTAC), Aug 2010
Final Remarks

- For graduate students:
  - Welcome research discussions to apply formal analysis techniques
    - Systematically testing/debugging C programs
    - Concurrency bug detection
    - Model-based testing
- Pre-requisite:
  - Basic understanding of the C programming language
  - Basic understanding of linux/unix