Realistic Verification Scenarios

Pavel Parízek

http://d3s.mff.cuni.cz
Concurrent HashMap (Java)

- Where to get it
  - `<JDK DIR>/src.zip:java/util/concurrent`
  - Alternative: [http://d3s.mff.cuni.cz/~parizek/teaching](http://d3s.mff.cuni.cz/~parizek/teaching)

- Task 1: identify relevant correctness properties
- Task 2: determine suitable verification approach
- Task 3: maybe actually use tools that you know
GNU core utilities

- Where to get the source code

- Selected target programs: cp, su

- Task 1: identify relevant correctness properties
- Task 2: determine suitable verification approach
- Task 3: maybe actually use tools that you know
HelenOS

- Source code
  - http://www.helenos.org/releases/HelenOS-0.5.0.tar.bz2

- Task 1: identify relevant correctness properties
- Task 2: determine suitable verification approach
- Task 3: how would you proceed with analysis of such large program
Interesting paper


- Key challenges
  - Parsing the real code (different standards of C/C++/Java)
  - Integration with the internally used build system
    - “Make” variants, not everybody uses Make, some GUI tools (IDE)
  - Bug reports: false positives, developers not understand
  - Social: convince managers that they want to fix bugs
• Competition on Software Verification

• http://sv-comp.sosy-lab.org/
• http://sv-comp.sosy-lab.org/2018/