Verification Tools in Practice

Testing, Quality Assurance, and Maintenance
Winter 2018

Prof. Arie Gurfinkel
How can one check a routine in the sense of making sure that it is right? The programmer should make a number of definite assertions which can be checked individually, and from which the correctness of the whole programme easily follows.
VerifyThis! Verification Competition

http://www.pm.inf.ethz.ch/research/verifythis.html

“Hackathon”-format

Given a few days and a few problems, create “provably” correct solutions

Solutions are judged by a committee to decide that they meet “provability” criteria.

Dafny is the most commonly used tools in the competition

Great way to learn about other tools, current research, and challenging problems
Microsoft Visual Studio Products

Code Contracts

- https://github.com/Microsoft/CodeContracts

- statically and dynamically checked method pre- and post-conditions

IntelliTest


- automated test generation by dynamic symbolic execution
Auto-active Verification Tools

Formal verification of code targeting developers

  - Functional PL approach: WhyML, VCGen, supports many SMT solvers
- **VeriFast**: [https://github.com/verifast/verifast](https://github.com/verifast/verifast)
  - concurrent and low-level memory, works directly on C language
  - standard for annotations for Java
  - based on ESC-Java by Rustan Leino (precursor of Dafny)
- **KeY**: [https://www.key-project.org/](https://www.key-project.org/)
  - Java, based on Symbolic Execution, found bugs in TimSort
- **Frama-C**: [https://frama-c.com/](https://frama-c.com/)
  - static analysis and auto-active verification for C
  - standard for annotations
  - Automation with static analysis, verification with Why3
Proving that Android’s, Java’s and Python’s sorting algorithm is broken (and showing how to fix it)

February 24, 2015  Envisage  Written by Stijn de Gouw.

Tim Peters developed the Timsort hybrid sorting algorithm in 2002. It is a clever combination of ideas from merge sort and insertion sort, and designed to perform well on real world data. TimSort was first developed for Python, but later ported to Java (where it appears as java.util.Collections.sort and java.util.Arrays.sort) by Joshua Bloch (the designer of Java Collections who also pointed out that most binary search algorithms were broken). TimSort is today used as the default sorting algorithm for Android SDK, Sun’s JDK and OpenJDK. Given the popularity of these platforms this means that the number of computers, cloud services and mobile phones that use TimSort for sorting is well into the billions.

General Purpose Theorem Provers

Logician view of program verification (and not just program)
ACL2: http://www.cs.utexas.edu/users/moore/acl2/
PVS: http://pvs.csl.sri.com/

HOL: https://hol-theorem-prover.org/
HOL-light: http://www.cl.cam.ac.uk/~jrh13/hol-light/
Isabelle: http://isabelle.in.tum.de/

Coq: https://coq.inria.fr/
  • https://softwarefoundations.cis.upenn.edu/plf-current/Hoare.html
  • Programs are extracted from proofs of theorems

Lean: https://leanprover.github.io/
Intermediate Languages for Verification

Build your own auto-active verifier for your favorite language (python, etherium, javascript, …)

BoogiePL: https://github.com/boogie-org/boogie
  • intermediate language of Dafny
  • “similar” to our while language

WhyML
  • Intermediate language of Why3

Viper
  • http://www.pm.inf.ethz.ch/research/viper.html
  • emphasis on concurrency and memory (inhale/exhale/implicity dyn. frames)

Constrained Horn Clauses
  • a fragment of FOL sufficient for invariants
  • used by some automatic tools
SPARKPro

http://www.adacore.com/sparkpro/

Ada is an old language still used in security- and safety-critical domains

SPARKPro provides auto-active verification for Ada

Actively used by specialized companies
IronClad and IronFleet

https://github.com/Microsoft/Ironclad

Distributed state machine and distributed key-value dictionary implemented in Dafny

Verified key correctness properties and key liveness properties
  • e.g., distributed algorithm eventually converges to a result as long as enough participants are active

Open sourced and available on GitHub

Non-trivial system designed and verified by system engineers
  • not researchers in FM trying to show off what their tools can do
Amazon S2N

Symbolic Execution Tools

KLEE: [https://klee.github.io/](https://klee.github.io/)
- EXE-style symbolic execution for LLVM virtual machine

S2E: [http://s2e.systems/](http://s2e.systems/)
- EXE-style symbolic execution at Virtual Machine Level

Angr: [http://angr.io/](http://angr.io/)
- A framework for symbolic execution and analysis of binaries

Mayhem: [https://forallsecure.com/blog/2016/02/09/unleashing-mayhem/](https://forallsecure.com/blog/2016/02/09/unleashing-mayhem/)
- Automated exploit generation using symbolic execution
S2E: In-vivo Analysis of Software Systems
Automatic Verification

Automatically verify specific properties of programs

- minimize user guidance
- fixed, not very deep properties

Microsoft Static Driver Verifier

- [https://docs.microsoft.com/en-us/windows-hardware/drivers/devtest/static-driver-verifier](https://docs.microsoft.microsoft.com/en-us/windows-hardware/drivers/devtest/static-driver-verifier)
- based on Model Checking
- checks for correct API usage rules: e.g., always lock-before-unlock

Linux Driver Verification: [https://linuxtesting.org/ldv](https://linuxtesting.org/ldv)

- Implementation of MS SDV for Linux using OSS tools

Facebook Infer: [http://fbinfer.com/](http://fbinfer.com/)

- Automatically prove correct memory handling (e.g., absence of null dereferencing) in C and Java programs
SeaHorn

A fully automated verification framework for LLVM-based languages.

http://seahorn.github.io
Smart Contracts: New Domain for Verification

Use code to represent a contract
Contract is executed by code
Blockchain is used to ensure adherence to the contract in a distributed decentralized environment

What does a contract actually do?
https://blockgeeks.com/guides/smart-contracts/
An option contract between parties is written as code into the blockchain. The individuals involved are anonymous, but the contract is the public ledger.

A triggering event like an expiration date and strike price is hit and the contract executes itself according to the coded terms.

Regulators can use the blockchain to understand the activity in the market while maintaining the privacy of individual actors’ positions.
The DAO Attack

https://medium.com/@MyPaoG/explaining-the-dao-exploit-for-beginners-in-solidity-80ee84f0d470

Follow the rules, use the contract, withdraw all money, get rich in the process
Symbolic Execution for Smart Contracts

MyThril
- [https://github.com/ConsenSys/mythril](https://github.com/ConsenSys/mythril)
- Security analysis tool for Ethereum Smart Contracts

Manticore: [https://github.com/trailofbits/manticore](https://github.com/trailofbits/manticore)
- Symbolic execution tool

- An analysis tool for smart contracts
- [https://oyente.melon.fund](https://oyente.melon.fund)

Very active area of both research and development.
Is Verification Enough

Can verified software fail?

Do we need both testing and verification?