WP verification methodology and tools

Paul Jackson

School of Informatics
University of Edinburgh

Formal Verification
Spring 2017

Levels of formal verification

- Checking freedom from run-time exceptions
  - Dominant level for SPARK tools
  - Not fully hands-off: typically need a few assertions (preconditions, postconditions, loop invariants, ...)
  - Might have some VCs needing checking by hand or by manually-guided proof in a proof assistant
- Property checking
  - Checking of critical properties that are relatively simple to express and generate VCs provable automatically
- Full checking of functional behaviour against specifications
  - Full automation possible for small programs, perhaps with assertion hints.
  - For larger programs and more complex properties, proof assistants needed. Proof by hand not tractable.

Use of assertions in run-time checking

Several benefits:

- Catches bugs during testing
- Gives programmers opportunity to gradually learn about and experiment with assertions
- Checks program inputs during tests conform to expectations
- Can check some complex properties that cannot be handled statically

Parallel story in digital hardware design world

Acceptance of assertions much higher than in software world

- Exist standardised LTL++ assertion languages
  - SVA SystemVerilog Assertions
  - PSL Property Specification Language
- Support from all standard commercial simulators
- Support also from formal and semi-formal commercial model checkers
- Integrated into both verification and design methodologies
  - Assertion Based Design

WP-based tools

Why3-based

Boogie-based

Others

Why3

Front-ends generating WhyML code and using Why3 tool:

gnat2why Used in Adacore and Altran's SPARK toolset.

Frama-C Platform for C formal verification.

- Includes WP plug-in for using Why3
- Other plug-ins for for flow analysis and test-case generation

Krakatoa For Java

Why3 language itself is human-friendly

- Examples library has over 100 textbook algorithms

Boogie

A intermediate-level verification language from Microsoft Research.

Front-ends include

- **Spec#** for C#
- **Dafny** Simple imperative language with heap data.
  - Popular in teaching
  - Recent application to secure web apps (Ironclad) and distributed systems (Ironfleet)
- **VCC** For low-level concurrent C.
  - Used to verify 60klines Hyper-V hypervisor.
- **SDV** Microsoft's Static Driver Verifier
  - Checks driver - Windows kernel interactions

Back-end analysis tools include:

- **Boogie tool** generates VCs for Z3 SMT solver.
- **Corral** Bounded loop unrolling – no use of invariants.
  - Used in SDV.

Other WP-based verification tools

Leon for Scala

OpenJML for Java.

- JML is Java Modelling Language, an assertion language
- Descendent of ESC/Java system