VCC: CONTRACT-BASED MODULAR VERIFICATION OF CONCURRENT C

Markus Dahlweid, Michal Moskal, Thomas Santen, Stephan Tobies
European Microsoft Innovation Center
Aachen, Germany

Wolfram Schulte
Microsoft Research, RiSE
Redmond, WA, USA

FEATURES:
- Correctness: Verified programs never go wrong (i.e., violate their specification)
- Concurrency: Operating systems stopped being single-threaded 20 years ago
- Modularity: Functions and data structures as natural abstraction boundaries
- Low-level C: Bit fields, unions, machine arithmetic, lock-free algorithms, ...
- Inline contracts: Specifications live and evolve along their code
- Tool integration: Plug into existing Microsoft developer tools
- Application: Used to verify Microsoft Hyper-V’s (virtualization) kernel

DATA STRUCTURE INVARIANTS:

WORKFLOW:
Annotate C code
Verify with VCC
Compile with regular C compiler

Analyze counterexample with Model Viewer
Analyze Z3 log with Z3 Axiom Profiler
Monitor proof search with Z3 Inspector

Fix code or specs with VCC VS plugin

FEATURES: 
- Correctness: Verified programs never go wrong (i.e., violate their specification)
- Concurrency: Operating systems stopped being single-threaded 20 years ago
- Modularity: Functions and data structures as natural abstraction boundaries
- Low-level C: Bit fields, unions, machine arithmetic, lock-free algorithms, ...
- Inline contracts: Specifications live and evolve along their code
- Tool integration: Plug into existing Microsoft developer tools
- Application: Used to verify Microsoft Hyper-V’s (virtualization) kernel

FUNCTION CONTRACTS WITH PRE- AND POSTCONDITIONS:

SPECIFICATION CODE TO (RE-)ESTABLISH INVARIANTS:

INTEGRATED ERROR REPORTING AND ANALYSIS:

STATE-OF-THE-ART AUTOMATIC SMT SOLVER
SMT (SATISFIABILITY MODULO THEORIES):
- Integer and fixed-length bitvector arithmetic
- Arrays, algebraic data types
Microsoft Research (L. de Moura, N. Bjørner)
E.g., also used by MSR’s Pex tool