| galois |

Proving Functional Correctness with the Software Analysis Workbench

Aaron Tomb, Galois

AFRL S5

August 2, 2017

What is SAW?

- SAW = Software Analysis Workbench
  - Software: many languages
  - Analysis: many types of analysis, focused on functionality
  - Workbench: flexible interface, supporting many goals
- What separates it from other systems?
  - One view: compiler :: imperative code → functional code
  - Captures all functional behavior, simplifying later if necessary
  - Uses efficient internal representations tuned to equivalence checking
  - Strong bit vector reasoning support
  - Focus on practicality over novelty
- Open source (BSD3) and available now

Current Capabilities

- Imperative to functional translation for programs that:
  - Have only bounded iteration
  - Have size-bounded inputs, outputs, and heap use
- Supports JVM, LLVM, and Cryptol (a DSL designed for cryptography)
- Translation to SAT or SMT to do things such as:
  - Prove two implementations equivalent (useful for regression verification)
  - Prove relations between inputs and outputs
  - Find inputs that lead to outputs with given properties
- Concrete execution and random testing
- User-guided rewriting

Example: Find First Set

```
uint32_t ffs_ref(uint32_t word) {
    if(!word) return 0;
    for(int c =  , i =  ; c <   ; c++)
        if(((  << i++) & word) !=  )
            return i;
    return 0;
}
```

```
uint32_t ffs_imp(uint32_t i) {
    char n =  ;
    if (!(i &  xffff)) { n +=   ; i >>=   ; }
    if (!(i &  x00ff)) { n +=   ; i >>=   ; }
    if (!(i &  x000f)) { n +=   ; i >>=   ; }
    if (!(i &  x0003)) { n +=   ; i >>=   ; }
    return (i) ? (n+((i+ ) &  x01)) :  ;
}
```

- Reference implementation
- Easy to understand
- Inefficient

- Optimized implementation
- Hard to understand
- Faster
- Identical functionality

Demo of FFS Verification

Verification of HMAC in s2n

- Amazon's open source implementation of TLS
- Verified C implementation of HMAC against Cryptol spec
  - ~15 LOC in Cryptol (high-level spec)
  - ~300 LOC in C (from official s2n repository)
  - ~400 LOC of script (all plumbing; likely smaller in the future)
- Proofs for various fixed message sizes
- Proof integrated into Travis build, running on every commit
  - Build and standard test: ~3-25 minutes
  - Proof for each hash algorithm: ~4-7 minutes
- Related: proof of OpenSSL's HMAC using VST (Coq)
  - Beringer, Petcher, Ye, Appel (2015)
  - SAW proof is more automated, less foundational

Demo of HMAC Integration

What Kinds of Code Work Well?

- Termination is guaranteed after a fixed number of iterations
  - SAW unrolls loops and recursion
- Inputs and outputs are fixed size (specific number of bits)
  - Pointers can exist, but layout needs to be known
- Code consists of pure computation
  - Any I/O or non-local control flow needs to be axiomatized
- A specification exists
  - Either a complete functional spec or a property
- You care about precise details of computations (bit-level reasoning)
  - Other tools more efficient for, e.g., proving memory safety
- Domains such as cryptography, serialization, signal processing

Summary

- SAW is a powerful tool for precise and flexible software analysis
  - Especially focused on equivalence checking
  - Supports automated reasoning with SAT/SMT
  - Uses Cryptol for high-level specifications
  - Can verify implementations in JVM, LLVM
- Focused on finite programs for now, with plans for generalization
- Many real-world case studies, mostly from cryptography
  - HMAC from s2n
  - BSM parser for V2V
  - AES from OpenSSL
  - Many others
- Available now as open source: https://saw.galois.com

Contributors

Aaron Tomb, Adam Foltzer, Adam Wick, Andrey Chudnov, Andy Gill, Benjamin Barenblat, Ben Jones, Brian Huffman, Brian Ledger, David Lazar, Dylan McNamee, Edward Yang, Eric Mertens, Fergus Henderson, Iavor Diatchki, Jeff Lewis, Jim Teisher, Joe Hendrix, Joe Hurd, Joe Kiniry, Joel Stanley, Joey Dodds, John Launchbury, John Matthews, Jonathan Daugherty, Kenneth Foner, Kyle Carter, Ledah Casburn, Lee Pike, Levent Erkök, Magnus Carlsson, Mark Shields, Mark Tullsen, Matt Sottile, Nathan Collins, Philip Weaver, Robert Dockins, Sally Browning, Sam Anklesaria, Sigbjørn Finne, Thomas Nordin, Trevor Elliott, Tristan Ravitch

Resources

- Cryptol
  - Web: https://cryptol.net
  - GitHub: https://github.com/GaloisInc/cryptol
- SAW
  - Web: https://saw.galois.com
  - GitHub: https://github.com/GaloisInc/saw-script
- HMAC verification
  - https://galois.com/blog/2016/09/verifying-s2n-hmac-with-saw/