An understanding on

CompCert: Formal verification of a realistic compiler

Krishnasree Kelath

The Problem before CompCert

- Safety-critical systems rely on source-level formal verification
- A miscompiling compiler breaks all your proofs
- Compiler becomes the weak link in safety-critical software
- Industry workaround: turn off optimisations, manually check assembly
- Need a compiler that guarantees correct behaviour of machine code

What is CompCert?

- Formally verified optimising compiler for a large subset of C (Clight)
- Implemented and proven correct in the Coq proof assistant
- Compiles Clight → PowerPC assembly through verified passes
- Semantic Preservation – Key Guarantee
- Designed for safety-critical embedded systems (avionics, medical, automotive)

How CompCert works

- Multi-pass Architecture: 14 compiler passes, 8 formally defined intermediate languages
- Each IR has a precise operational semantics
- Every pass is proven to preserve semantics
- End-to-end correctness follows by composition

Core Concepts

- Semantic preservation (CompCert’s main theorem)
- Formal operational semantics
- Verified multi-pass compilation
- Validators for complex optimisations
- Data Flow Analysis

Performance of CompCert

- Runtime performance is within 7–12% of GCC -O1/-O2, far outperforming GCC -O0
- Lightweight, verified optimisations (constant propagation, CSE, register allocation) still produce high-quality code
- Demonstrates that formal verification does not require sacrificing efficiency
- Although performance is not the main aim, results show that correctness can coexist with practicality

Pros and Cons of CompCert

Pros:

- End-to-end machine-checked correctness
- Removes compiler from “weak link” in verification chain
- Competitive performance: near GCC -O1/-O2
- Clear semantics for each IR; proofs expose subtle bugs early

Cons:

- Supports only Clight (subset of C)
- Targets PowerPC only in this version
- High verification effort (tens of thousands of Coq lines)
- Parser, printer, assembler/linker remain unverified

Conclusion

Award winning CompCert bridges formal proof and practical compilation, ensuring reliable machine code with proven correctness.