

Programming Language (4190.310)

Chung-Kil Hur (허충길)

Department of Computer Science and Engineering
Seoul National University

What we learn in the course.

➤ Interactive Theorem Proving in Coq

- Mathematical Logic
- Set Theory
- Functional Programming

➤ Imperative Programming Language

- Type Theory
- Semantics
- Program Verification

My passion towards rigorous math

My question when I was young:

Can we do math rigorously?

In a set theory course:

Yes! We can do math only using first-order logic with ZFC.

But, I was disappointed:

because it looked practically impossible to do math in such a way.

Formal math thanks to computers

Now I do math formally using Coq:

Computers do such tedious details for me!

I will talk about Coq today.

What Is A Proof Assistant?

- **Underlying Logic** for constructing Propositions & Proofs
- **Set theory** in the Logic for defining Sets & Elements
- **Tool** that implements such a logic and a set theory
- **Independent Proof Checker** that checks validity of given definitions and proofs

Examples of Proof Assistants

➤ Conventional Mathematics

- First-order logic
- Zermelo–Fraenkel set theory with the axiom of choice (ZFC)
- No Mechanization

➤ Isabelle/HOL

- Higher-order logic
- Function Space + Inductive Set
- Tool and Proof Checker
- Developed at University of Cambridge & Technical University of Munich

➤ Coq

- Calculus of Construction (Logic = Set Theory = Programming Language)
- Tool and Proof Checker
- Developed at INRIA, France

Demo of Coq

Set Theory

= Logic

= Functional Programming

Applications of Proof Assistants

➤ 4-Color Theorem & Feit-Thompson theorem

- Any map in a plane can be colored using four-colors.
- Every finite group of odd order is solvable.
- Mechanized in Coq by Georges Gonthier.
- (Note) Kenneth Appel and Andrew Appel.

➤ seL4 (secure embedded L4)

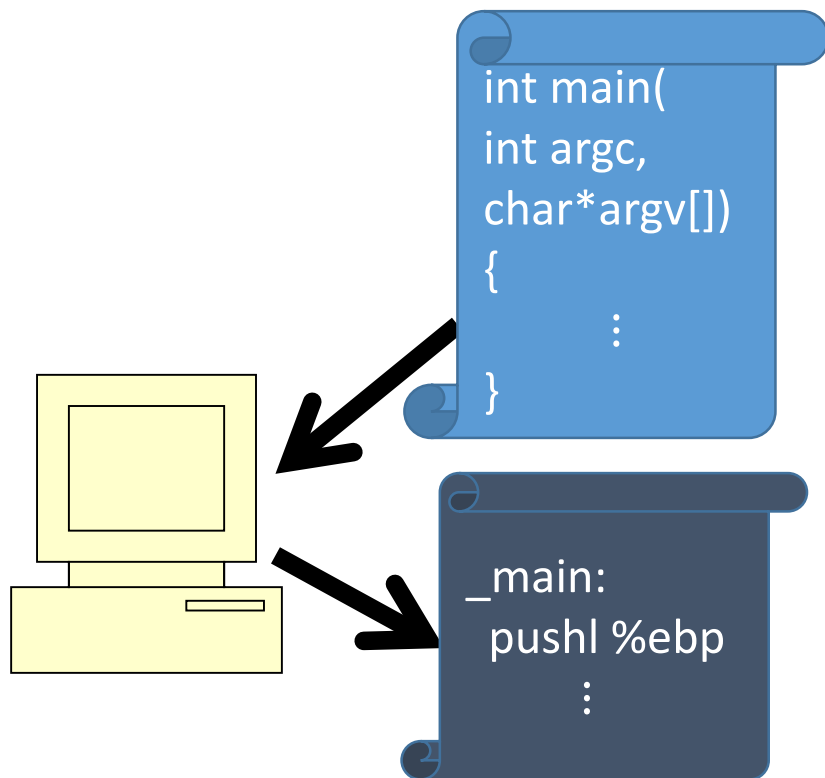
- Fully verified highly-optimized Microkernel based on L4
- at NICTA, Isabelle/HOL

➤ CompCert

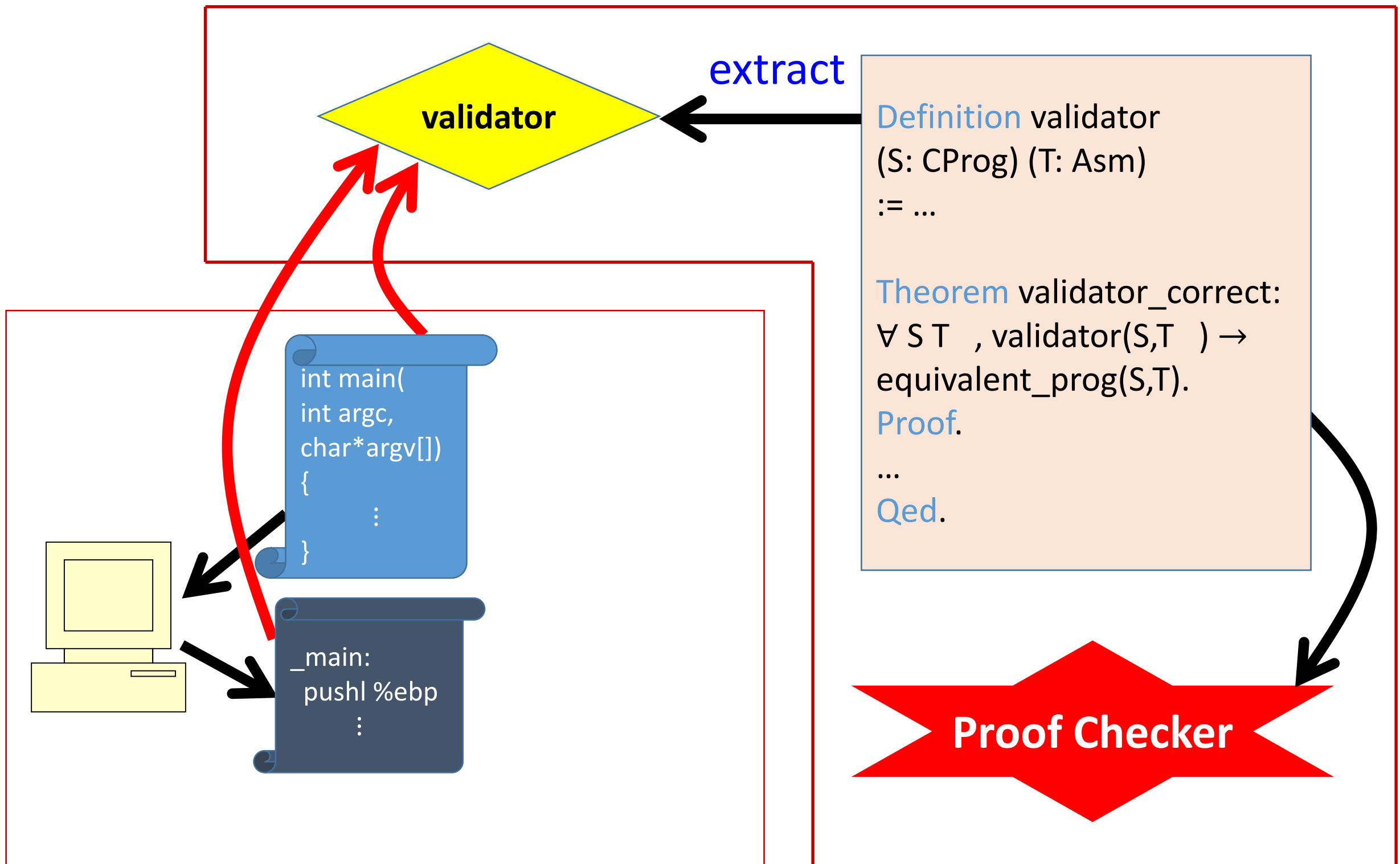
- Fully verified optimizing C compiler
- at INRIA, Coq (by Xavier Leroy)
- 79 bugs in GCC, 202 bugs in LLVM, **no bugs** in CompCert
- Its use in Airbus is being investigated.

➤ Security Protocols, Homotopy Type Theory, ...

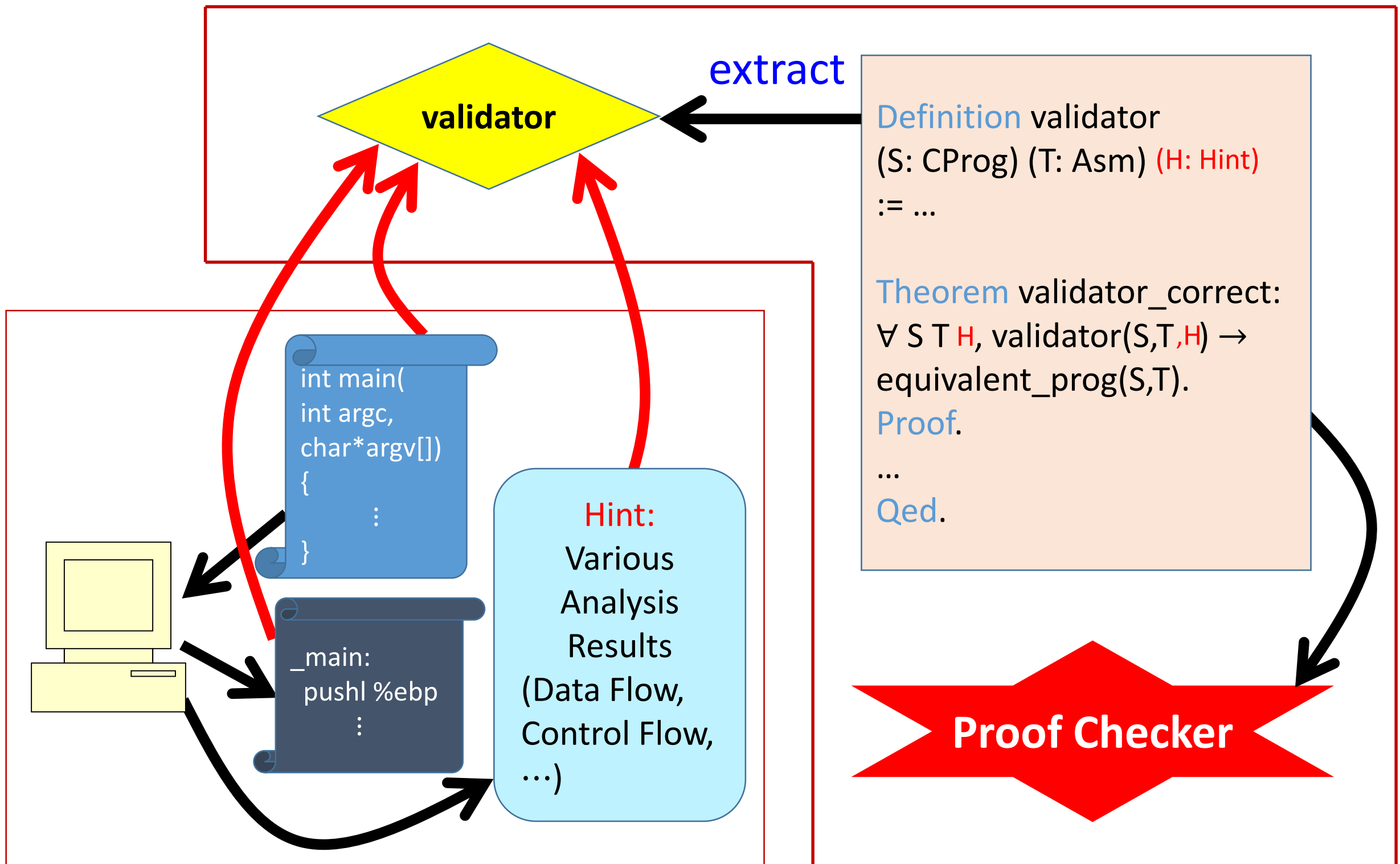
My Research: Compilation Validation



My Research: Compilation Validation



My Research: Compilation Validation



My Research: Compilation Validation

Validators can guarantee **absence of bugs**.

We are currently developing
a **Verified Validator for LLVM Compiler**.

