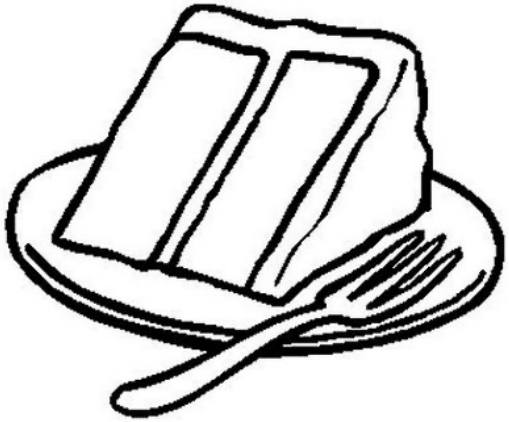
Have Your Verified Compiler And Extend It Too



Zachary Tatlock
Sorin Lerner
UC San Diego

Compiler Correctness

Building robust compilers is difficult

complex interactions resist testing

Compiler bugs are contagious

invalidate source level guarantees

Few users extend their compiler

hand optimized, unreadable code

Verified Compilers

- Implement compiler in proof assistant
- Prove compiler correct interactively
- CompCert [Leroy], Lambda Tamer [Chlipala]

Strong Guarantee

Difficult to Extend

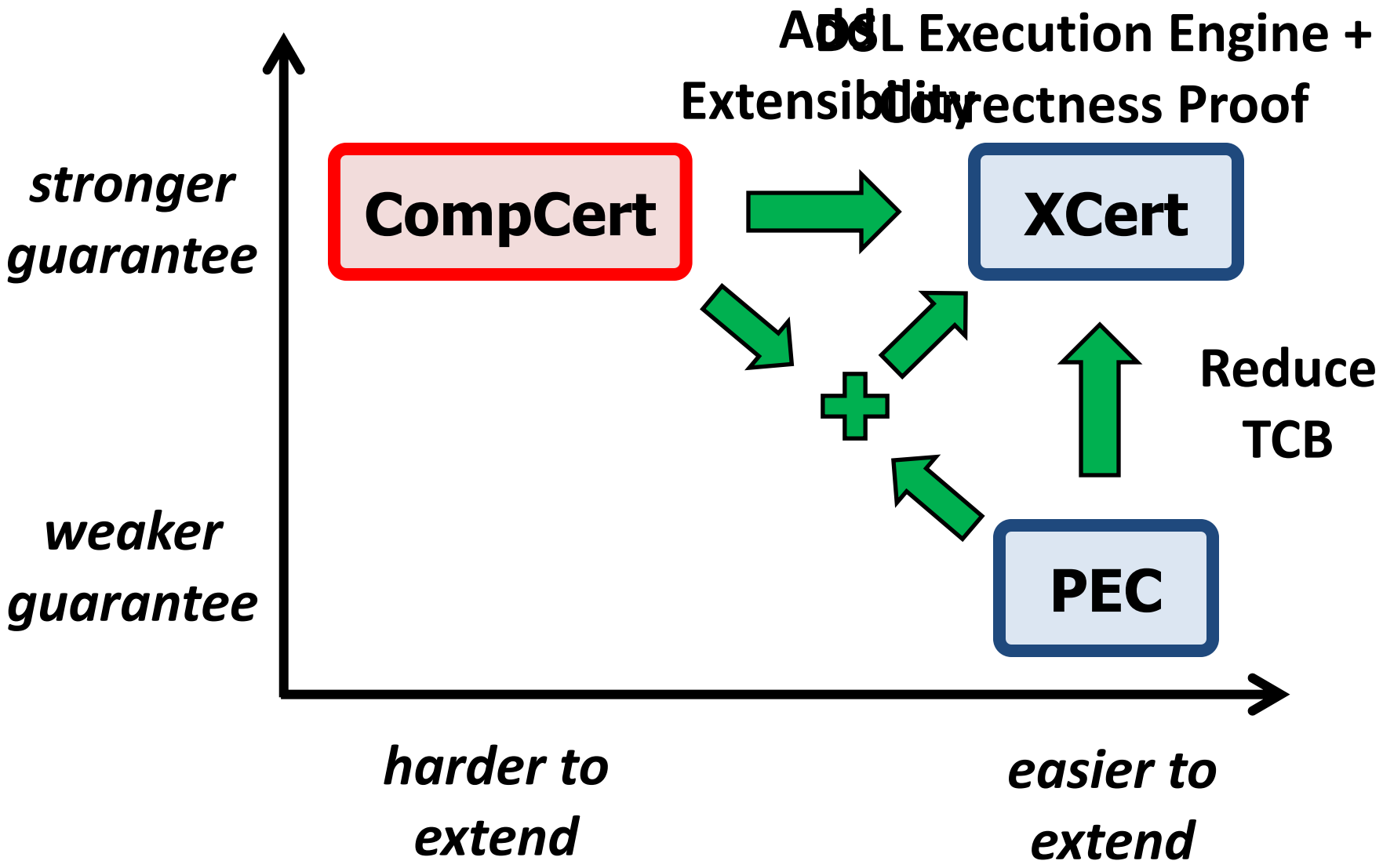
DSL-based Compilers

- Domain Specific Language for optimizations
- DSL opts proven correct automatically
- Rhodium [POPL 05], PEC [PLDI 09]

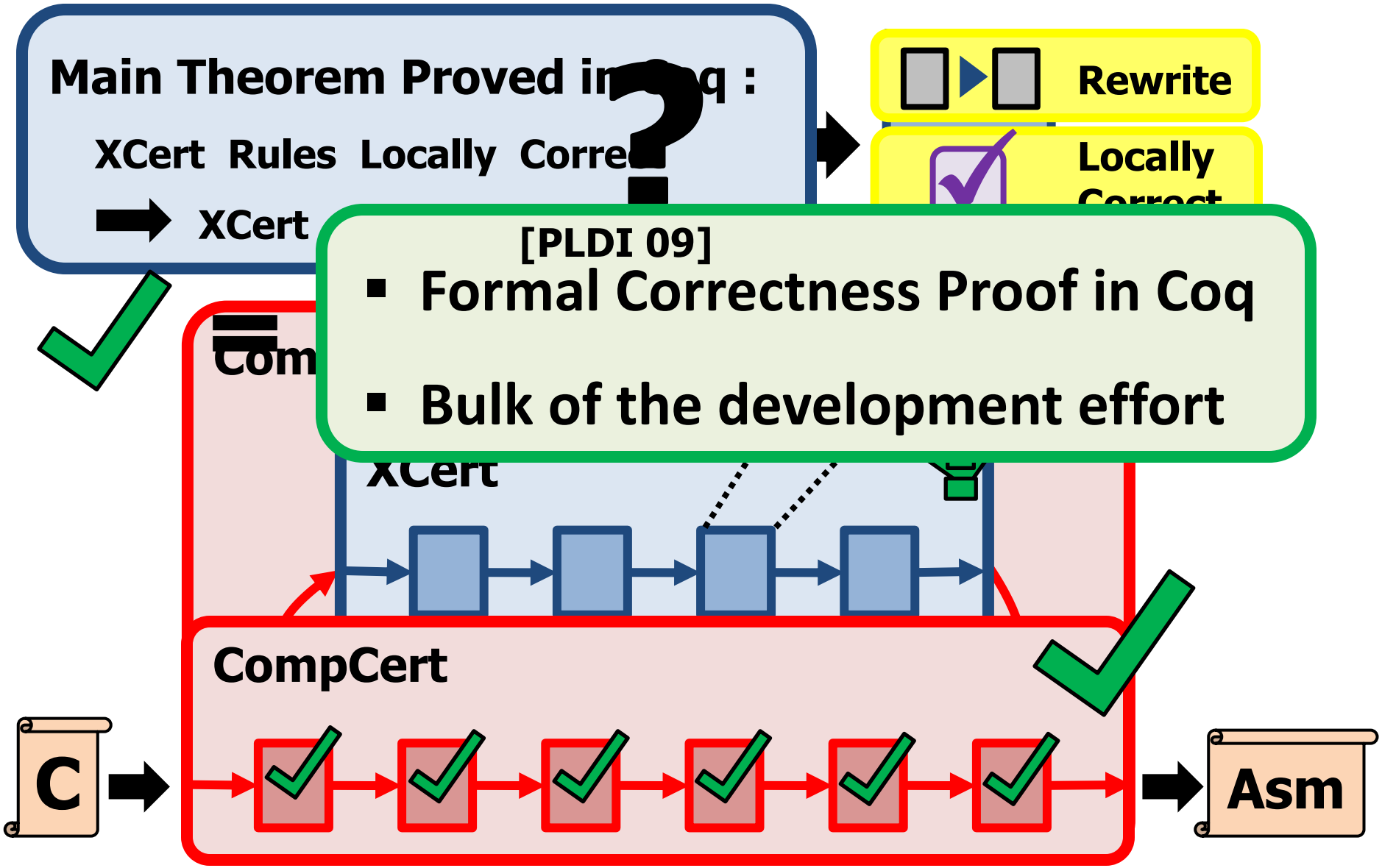
Easier to Extend

Weaker Guarantee

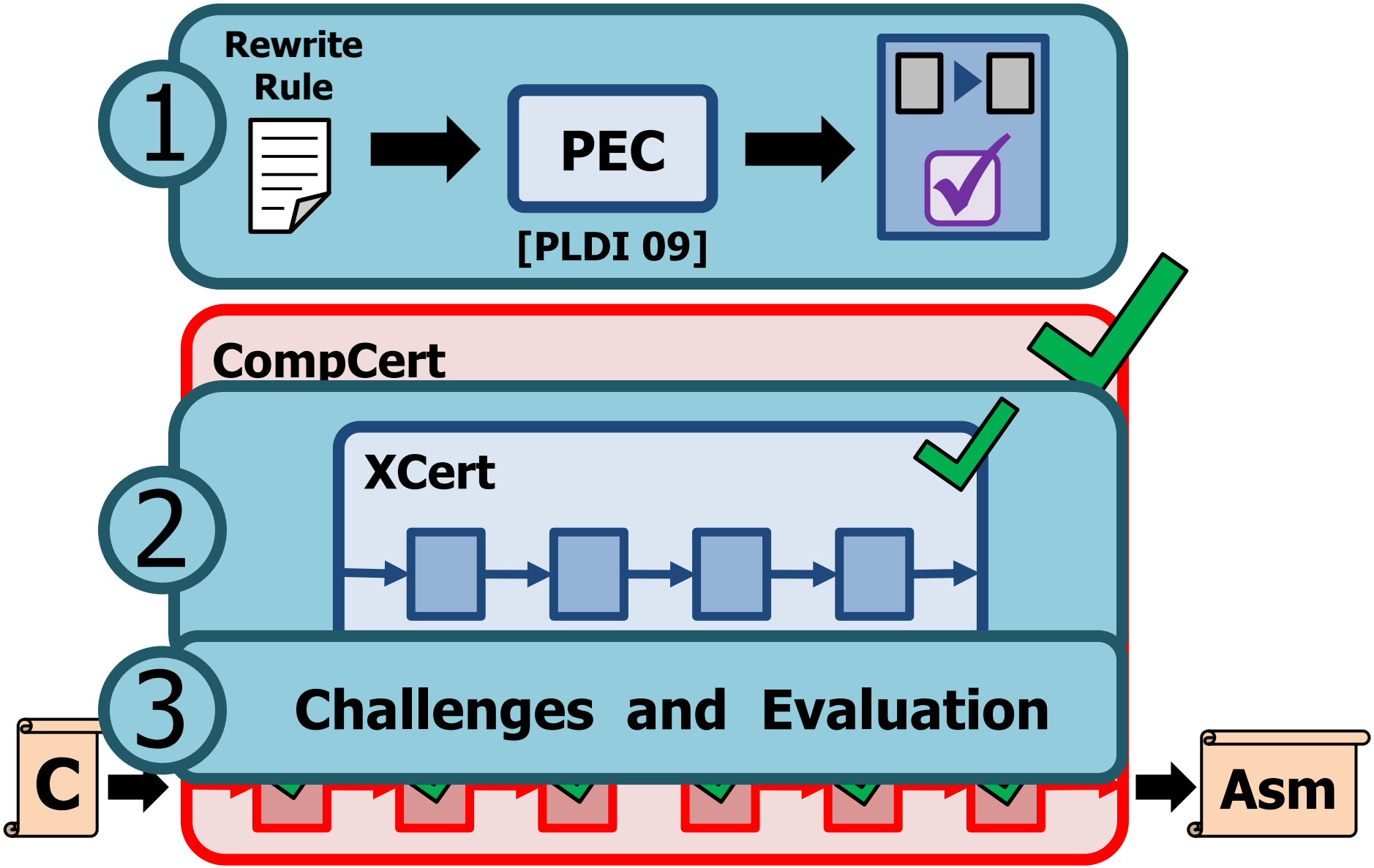
Contribution



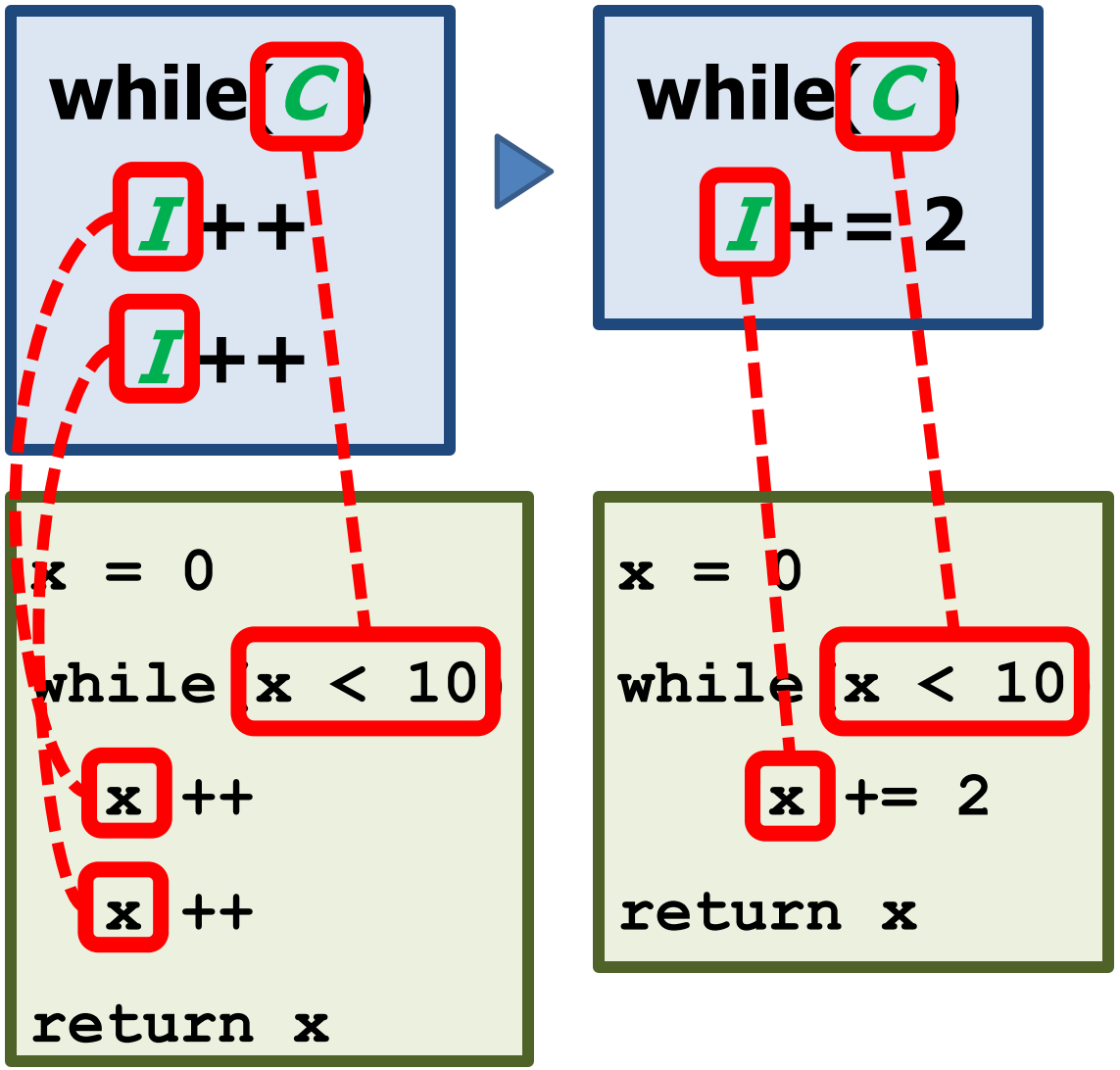
Extensible & Certified Compiler



Extensible & Correct Compiler



[PLDI 09]



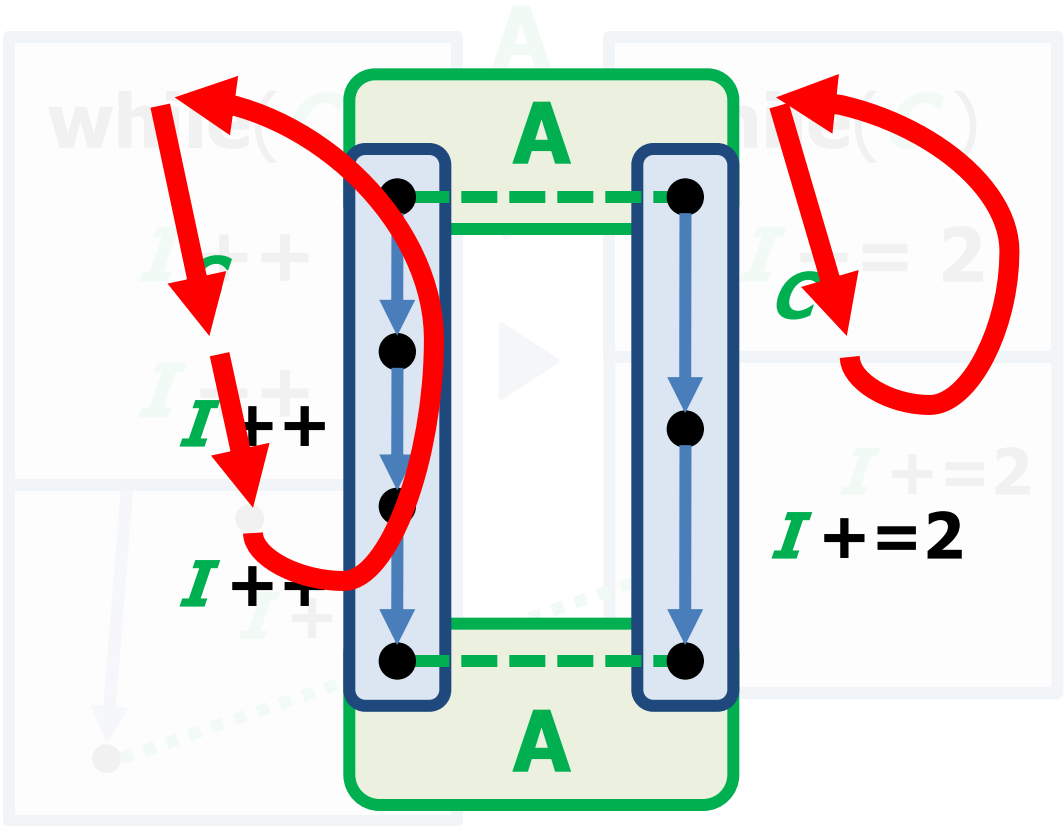
Rewrite Rule

- Find & Replace
- Match Pattern

  C → x < 10

  I → x
- Apply Subst

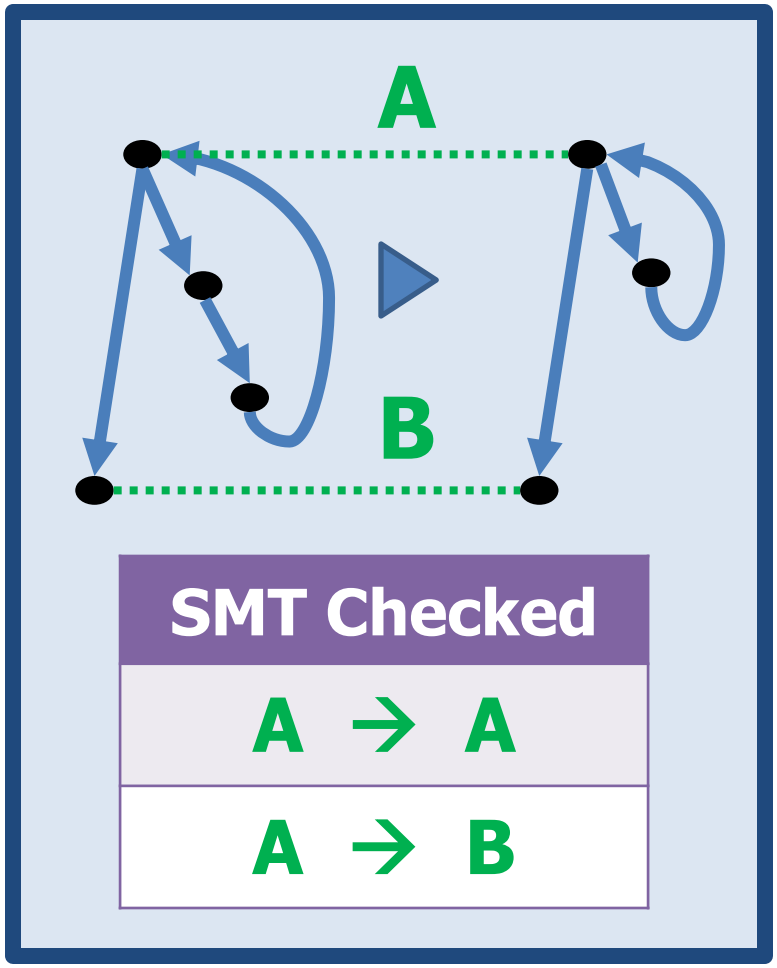
[PLDI 09]



PEC Checker

1. Convert to CFG
2. Guess Sync Points
3. Check w/ SMT

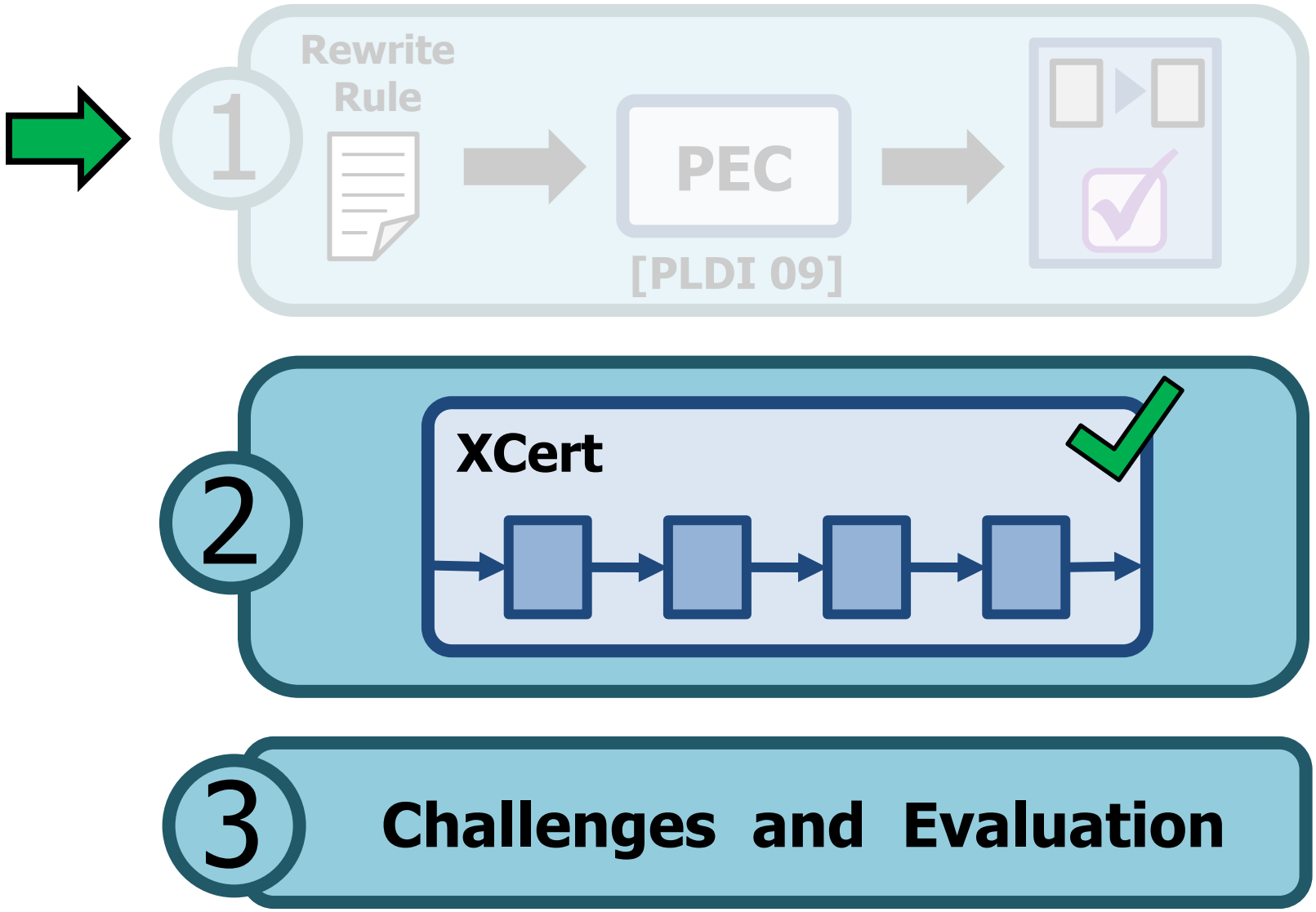
SMT Checked
A → A ✓
A → B ✓



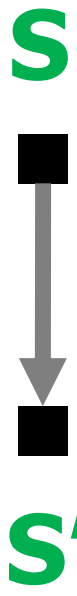
XCert Module

1. Rule in Coq
2. SMT Checks

Extensible & Correct Compiler



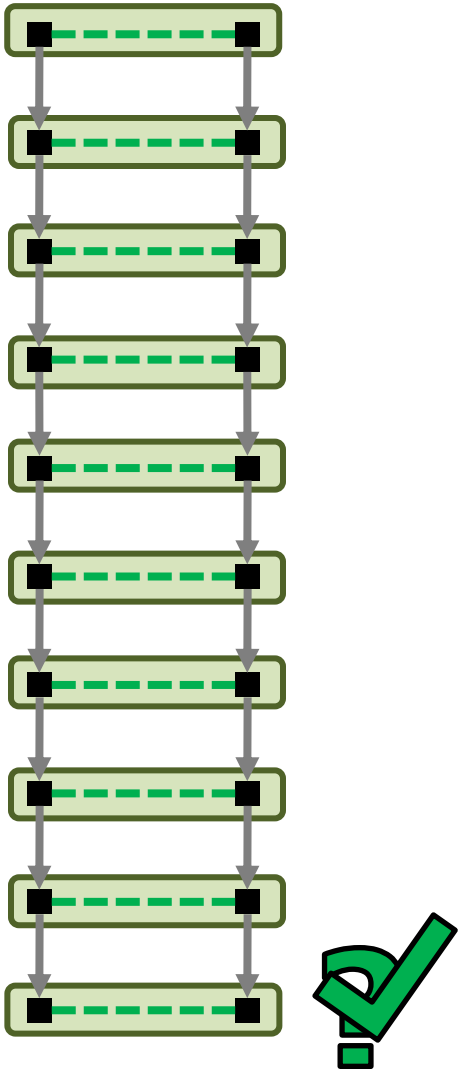
XCert Correctness Proof



Small Step

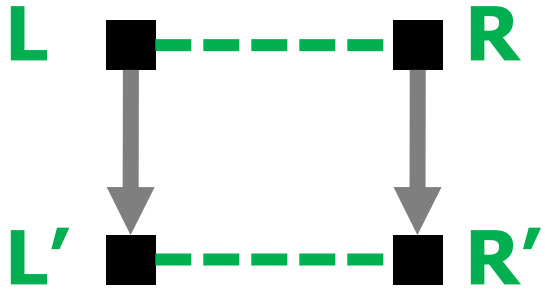
- Execute instruction
- Step state **S** to **S'**

XCert Correctness Proof



Equivalent Executions

- Initial Equiv ➡ Final Equiv
- Prove Simulation Diagram

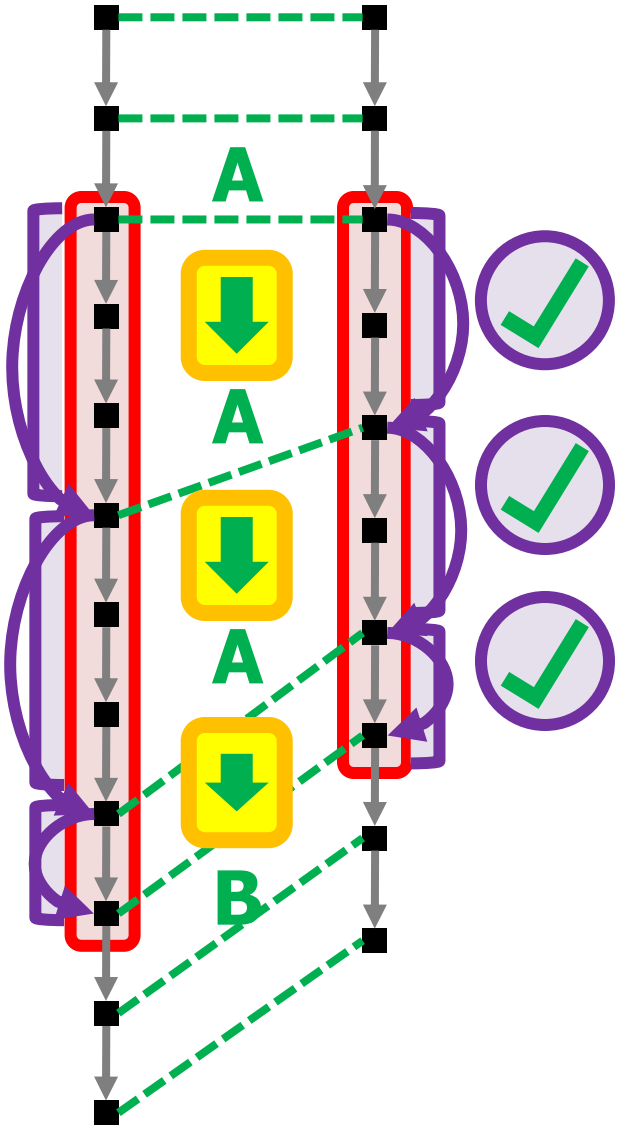


- CompCert Small Step Library:

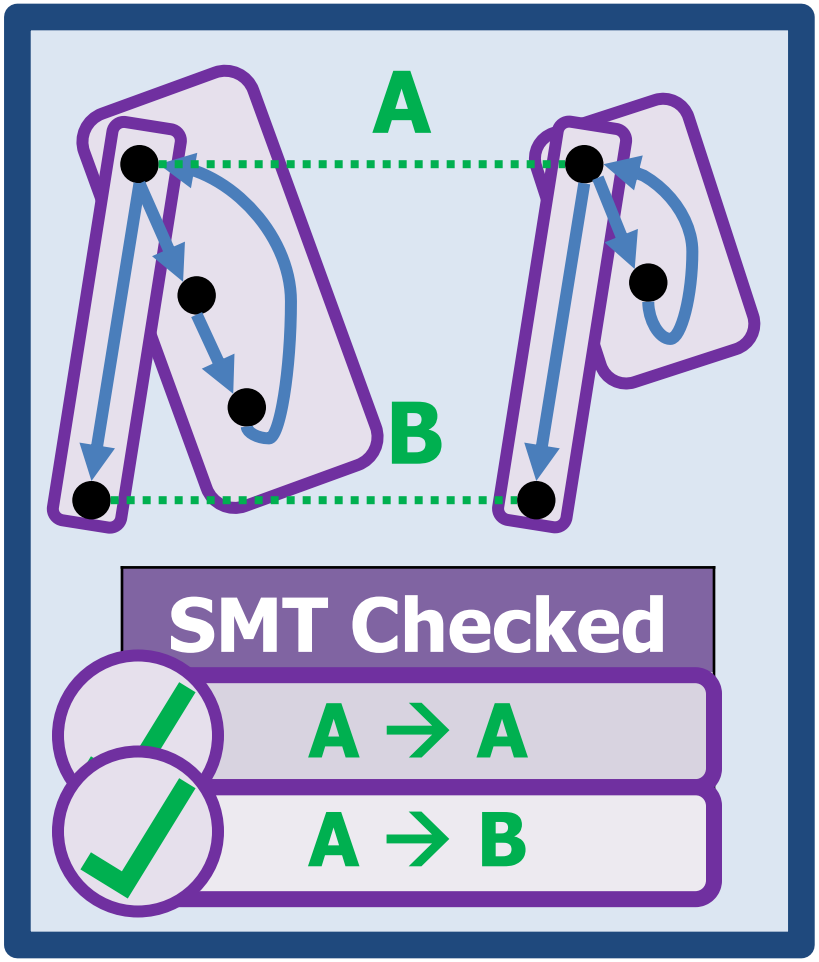
$$\frac{L \sim R \quad \wedge \quad L \Rightarrow L'}{\text{Sim Diagram} \quad \Rightarrow \quad \text{Progs Equiv}}$$

$$\exists R' : R \Rightarrow R' \quad \wedge \quad L' \sim R'$$

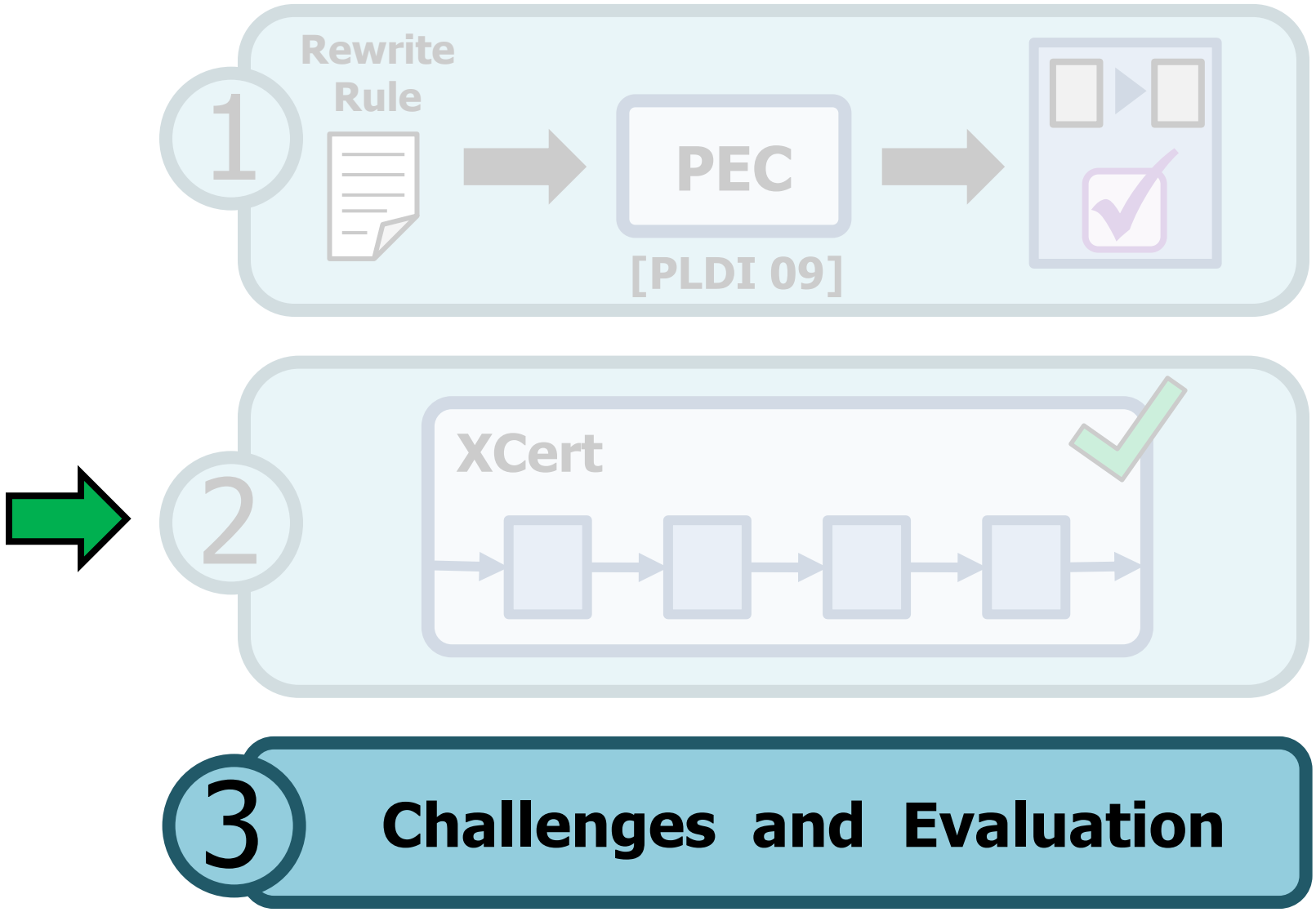
XCert Simulation Diagram



XCert Module



Extensible & Correct Compiler



Challenges (see paper)

XCert Execution Engine

- CFG pattern matching
- CFG splicing

XCert Correctness Proof

- Managing case explosion
- Verified validation **[Tristan and Leroy]**
- Preserving non-terminating behaviors

Evaluation

Engine : 1,000 lines of Coq functional code

Proof : 3,000 lines of Coq proof script

Trusted Computing Base (TCB)

- Compcert : Coq + Coq encoding of C sem
- XCert adds : SMT + SMT encoding of C sem

Evaluation

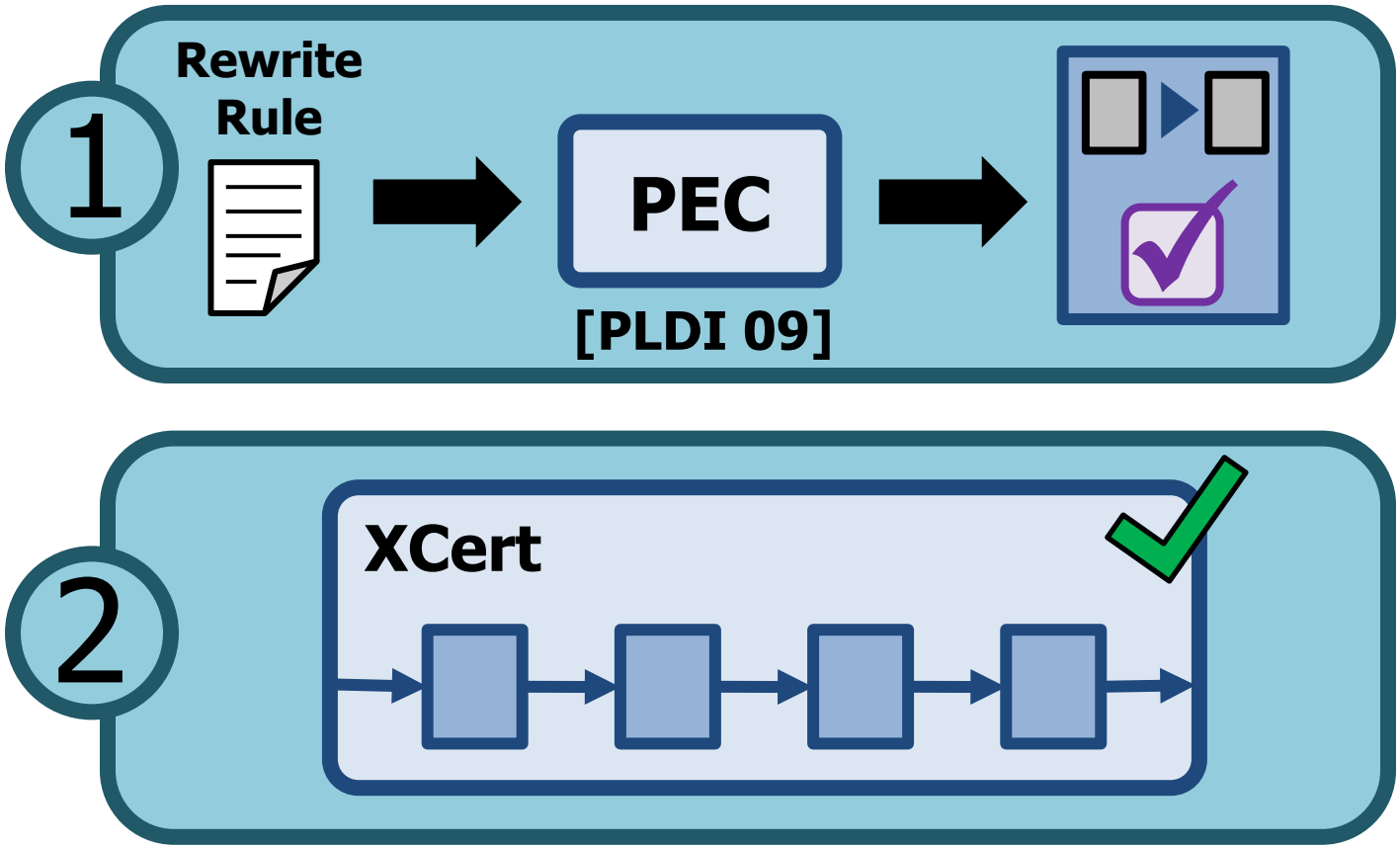
Extensibility: Support PEC Opts **[PLDI 09]**

- No manual proof effort or TCB increase
- Maintain Compcert end-to-end correctness

Sample of Optimizations Run:

Loop Invariant Code Hoist	**Loop Peeling**
Software Pipelining	**Conditional Speculation**
Loop Unswitching	**Partial Redundancy Elim**

Extensible & Correct Compiler



Thank You!