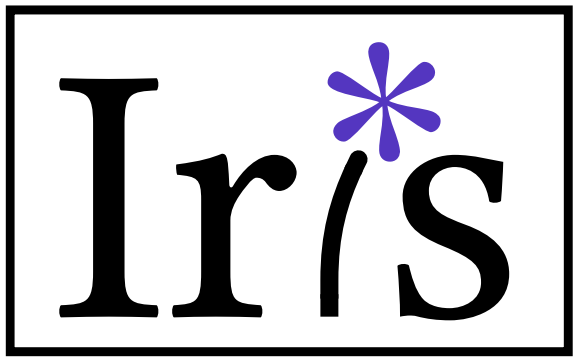
Questions

Nobody has responded yet.

Hang tight! Responses are coming in.

Iris Projects

- Logical atomicity
- Weak memory (iGPS, iRC11)
- Distributed systems (Aneris, Actris)
- Later credits
- Transfinite step-indexing
- …



- Rust (RustBelt)
- Go (Perennial)
- C (RefinedC)
- WebAssembly (Iris-Wasm)

HeapLang + extensions

real languages

How do you verify a real program in Iris?

1. Translate the program into a *model* in Coq
   - Could be a HeapLang-like language, or another kind of datatype
2. State a spec for each function
   - Details can vary, but points-to, invariants, ghost state, etc. mostly the same in all languages
3. Prove the specs using wp tactics for that language
   - iDestruct, iApply, etc. are universal, but wp_ are per-language

- Exercise: What's a feature of real languages that HeapLang doesn't have?

How do you verify a real program in Iris?

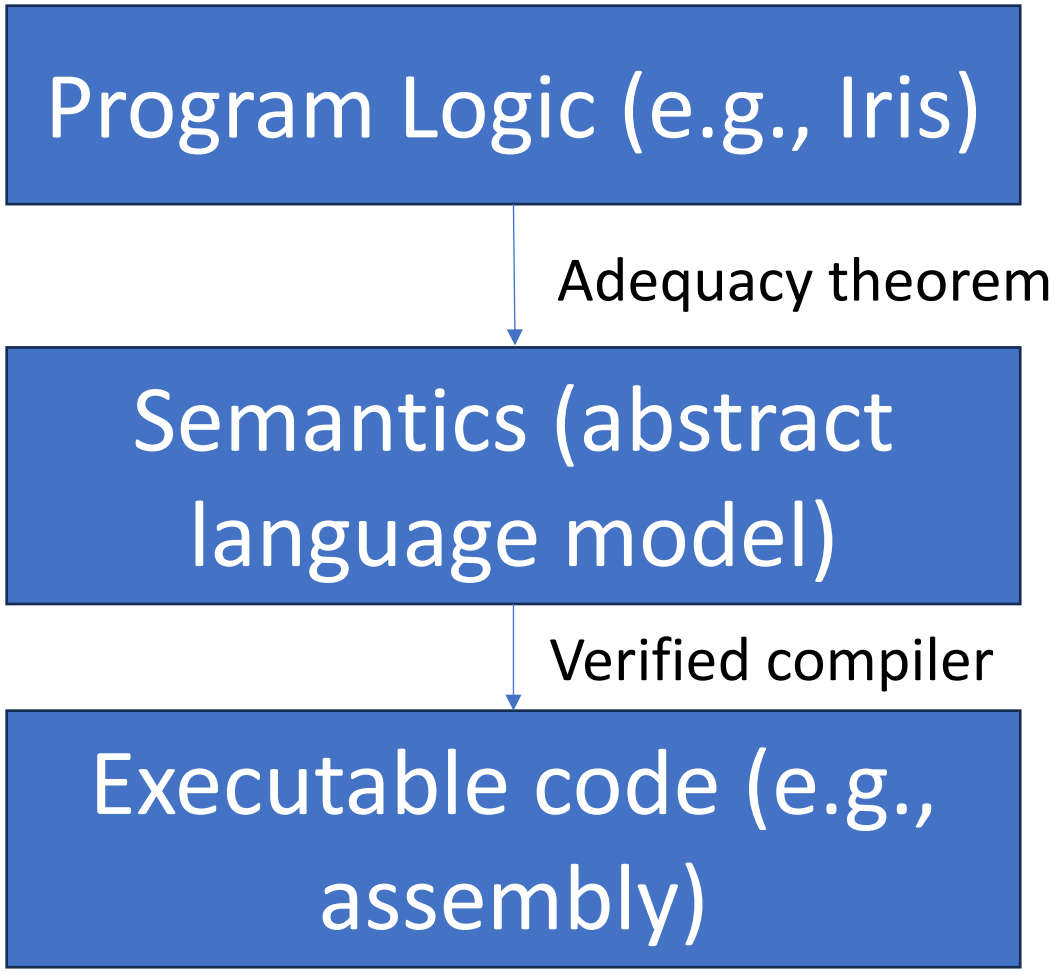
1. Translate the program into a *model* in Coq
   - Could be a HeapLang-like language, or another kind of datatype
2. State a spec for each function
   - Details can vary, but points-to, invariants, ghost state, etc. mostly the same in all languages
3. Prove the specs using wp tactics for that language
   - iDestruct, iApply, etc. are universal, but wp_ are per-language

4. Using the *adequacy* theorem, know that the verified program actually does the right thing!

Some Iris instances for real languages

- lambda-rust (Rust): https://gitlab.mpi-sws.org/iris/lambda-rust
- Perennial (Go): https://github.com/mit-pdos/perennial
- Iris-Wasm (WebAssembly): https://github.com/logsem/iriswasm
- RefinedC (C): https://gitlab.mpi-sws.org/iris/refinedc

What can we know about a verified program?



Separation logics with verified compilers

- Verified Software Toolchain (C): https://github.com/PrincetonUniversity/VST/, see also Software Foundations Volume 5

- CakeML (SML): https://cakeml.org/index.html