FORMAL LAND

BETTER RUST FORMAL VERIFICATION

Improvements we are making to coq-of-rust to verify Revm

MARCH 2025

DEV OF LINKING

- Phase to add typing and name resolution
- Next phase will be to handle memory

WHAT IS HARD

- Some files must be cut for mutual dependencies
- We must handle Rust’s standard library
- Some types are complex, like traits with associated types

AUTOMATION

- Generation in Python of Rocq code for type definitions
- Link: conversion to concrete types/values
- of_ty: conversion from concrete types
- of_value_with/of_value: conversion from concrete values, knowing of not the type
- Sub-pointer lens

TACTIC

- Tactic “run_symbolic” to automatically apply the reasoning rules for each node of the AST
- Can solve a lot of cases but still slow, and hence not fully automatic

SOLUTION

- Using more type classes instead of the **simple** tactic
  - Mid-term: use **export** visibility to avoid polluting the current context
- Applying tactic only on visible constructors: we tried, but gains are not clear and it might even be more complex in some cases
- Having all tactics that apply to constructors: requires changing the free monad of the translation. It would make things more uniform, maybe faster, as rewrites can be slow/fragile

NEXT

Have a layer to handle the memory

THANKS

 contact@formal.land