

Título: Modelado Formal de Rust y Compilación Verificada Fuentes Principales: "HACSPEC.pdf", "Fiat-crypto ERBSEN.pdf", "CompCert.pdf"

Resumen Técnico Denso: El objetivo es optimizar código Rust. Para que Lean pueda razonar sobre él, se debe utilizar un subconjunto formalizable como Hacspect. Hacspect es un subconjunto puramente funcional de Rust diseñado para especificaciones criptográficas, que evita el borrowing mutable y utiliza el sistema de tipos de Rust para modelar restricciones de dominio.

El flujo de trabajo propuesto imita la arquitectura de Fiat Cryptography:

1. Especificación: Código de alto nivel en un DSL dentro del asistente de pruebas (Lean en este caso, Coq en Fiat).
2. Optimización: Aplicación de reglas de reescritura verificadas para especializar la aritmética (ej. reducción modular, optimizaciones de curvas elípticas).
3. Generación: Emisión de código de bajo nivel (Rust/C) que es correcto por construcción, eliminando la necesidad de verificación a posteriori del ejecutable generado.

Es crucial distinguir entre la verificación semántica (como en CompCert, que prueba la preservación de comportamiento traza a traza) y la síntesis de programas (como en Fiat, que genera código desde especificaciones). Este proyecto busca lo segundo: un optimizador que sintetiza versiones eficientes de Rust explotando propiedades matemáticas (anillos, cuerpos finitos) que compiladores tradicionales como LLVM no pueden ver.