

Título: Arquitectura de Metaprogramación y Verificación en Lean 4
Fuentes Principales: "MetaM in lean4.pdf", "Comprehensive survey of Lean 4...pdf"

Resumen Técnico Denso: El proyecto se basa en la capacidad de Lean 4 para acceder a sus estructuras de datos internas mediante el paquete Lean, permitiendo la implementación de automatización de pruebas compilada eficientemente en C. La metaprogramación se organiza alrededor de mónadas jerárquicas: CoreM (acceso al entorno), MetaM (contexto de metavariables y unificación), TermElabM (información de elaboración) y TacticM (gestión de objetivos).

La manipulación sintáctica es crítica para el "frontend" del optimizador. Lean convierte el código (String) en objetos Syntax y luego en Expr (Expresiones). Las expresiones (Expr) son el núcleo, incluyendo constructores como Expr.lam (abstracción lambda), Expr.const (constantes globales) y Expr.app (aplicación de funciones). La elaboración transforma la sintaxis concreta en estas expresiones abstractas tipadas, manejando la inferencia de tipos implícitos y la instanciación de metavariables.

Para la optimización matemática, es vital el uso de MetaM para manipular el contexto local (LocalContext) y realizar reducciones (reduce) o verificaciones de igualdad definicional (isDefEq). La jerarquía algebraica de Mathlib (Semigrupo → Anillo → Cuerpo) proporciona la semántica para las reglas de reescritura.

Conceptos Clave para el Optimizador:

- Reflexión: La capacidad de Lean de definir sintaxis personalizada y rutinas de elaboración en el mismo entorno.
- Variables de De Bruijn: Representación de variables ligadas (bvar) en Expr para evitar colisiones de nombres durante la manipulación de términos.
- Transparencia: Control sobre qué constantes se despliegan durante la reducción (TransparencyMode), crucial para optimizar el rendimiento del compilador.