

Título: Algoritmos Algebraicos para Optimización de Código Fuentes Principales: "Computational Linear Algebra", "Algebraic Machine Learning", "Arithmetic Circuits Survey"

Resumen Técnico Denso: El optimizador debe reconocer estructuras algebraicas en el código Rust fuente para aplicar algoritmos asintóticamente más rápidos.

- Álgebra Lineal: Implementar transformaciones que mapeen bucles anidados a operaciones matriciales sobre cuerpos finitos. Utilizar algoritmos como la eliminación Gaussiana recursiva o multiplicación rápida de matrices (Strassen/Winograd) cuando se detecten patrones compatibles.
- Circuitos Aritméticos: Modelar el cómputo como circuitos aritméticos (DAGs de sumas y productos). El análisis de profundidad y tamaño de estos circuitos permite aplicar lemas de reducción de grado y homogeneización.
- Machine Learning Algebraico: Utilizar estructuras de semireticulo (semilattice) y descomposición de átomos para encontrar representaciones mínimas de funciones complejas, lo cual puede guiar la búsqueda heurística en el espacio de optimización.

La biblioteca Mathlib de Lean provee las definiciones formales (Anillos, Cuerpos, Espacios Vectoriales) necesarias para probar que estas transformaciones (ej. $A \cdot B \rightarrow \text{Winograd}(A, B)$) preservan la semántica denotacional del programa original.