Alpha-Equivalence Analysis via De Bruijn Indices

Antigravity Agent

January 29, 2026

1 Methodology

We applied lambda calculus-inspired techniques to detect Common Subexpressions (CSE):

- **De Bruijn Indices**: Variable names normalized to binding distance
- **Hash Consing**: DAG canonicalization for structural equality
- **Tree Pattern Matching**: Subtree extraction and comparison

2 Common Patterns Detected

The following patterns (modulo alpha-equivalence) appear in multiple locations:

Pattern Hash	**Occurrences**
43425b64ca35...	30
292075a08b38...	33
f06b4bafd3e6...	9
7ce493cf3e90...	12
598af978dc54...	10
dc2c6c712030...	7
f419212f408f...	10
1808aa08862d...	8
19244e041e5c...	8
8bdc70be4782...	8
a4c390856931...	8
14317903449f...	7
fab5e31572d4...	7
6694aeeeee47...	5
9ea622762d11...	6

3 Interpretation

Patterns with high occurrence counts indicate opportunities for refactoring via Common Subexpression Elimination (CSE). The use of De Bruijn indices ensures we detect equivalences even when variable names differ.