Implementation of the TIL-Script Language

Filip Peterek

VSB – Technical University of Ostrava

May 2023

Project Goals

- Define the TIL-Script programming language
- Create a working TIL-Script interpreter
- Document the language properly

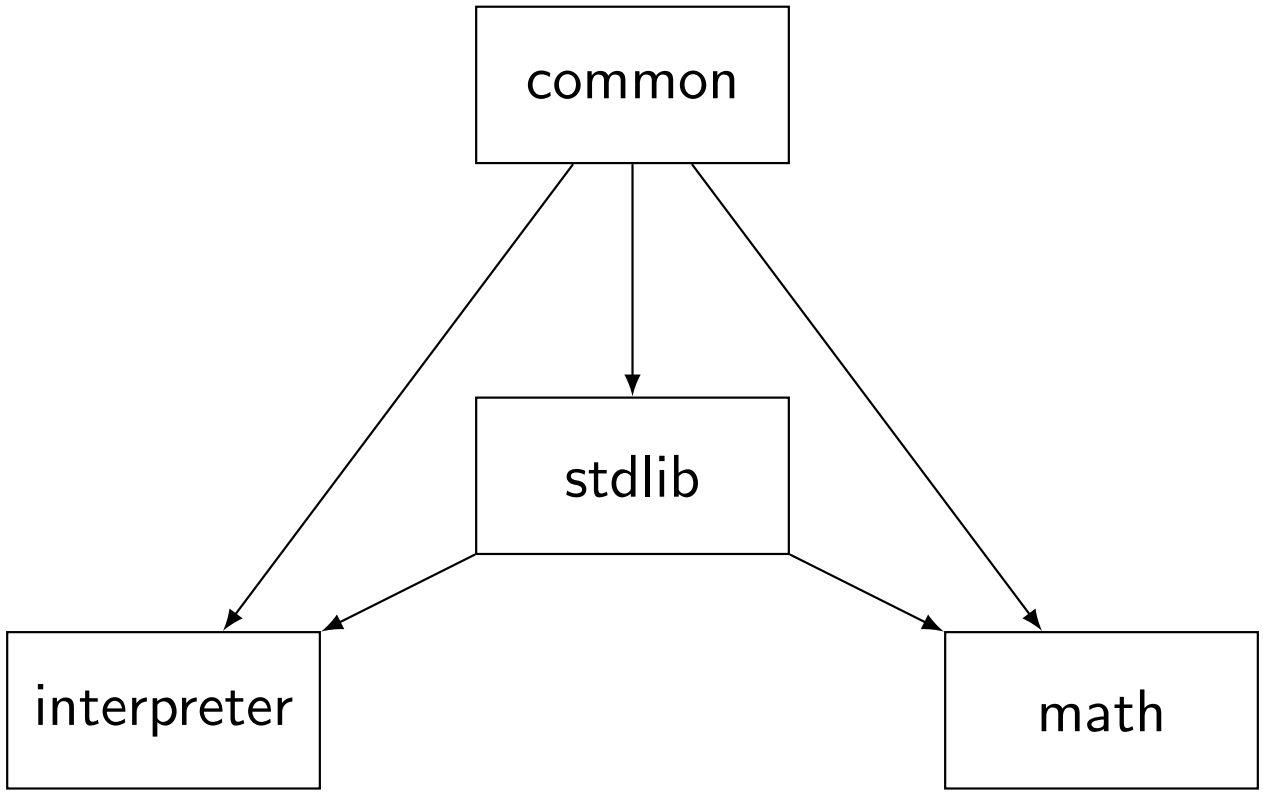
Transparent Intensional Logic

- Logical analysis of natural language
- Based on typed Lambda calculi
- Rigorously defined type hierarchy
- Procedural and hyperintensional
- Constructions can mention other constructions
- Sentence meaning is carried by a procedure
  - We mostly care for the procedure itself, seldom do we care for the value it produces

Implementation

- Kotlin
- Gradle
- Antlr

Project Structure



Parser Implementation

- Existing TIL-Script grammar has been utilized
  - Slight tweaks were necessary to make it work with Antlr
- Parser is generated using Antlr
- Resulting AST is converted to a custom structure

Typechecker Implementation

- Recursive implementation
- Aliases are expanded, expansions are compared
- Double execution is ignored
- Trivializations force construction of independent scopes
- Type deduction is unsupported
- Errors are, yet again, printed to stdout

Context Recognition

- Context recognition is also supported
- Recursive implementation
- No errors can be detected at this point

The End