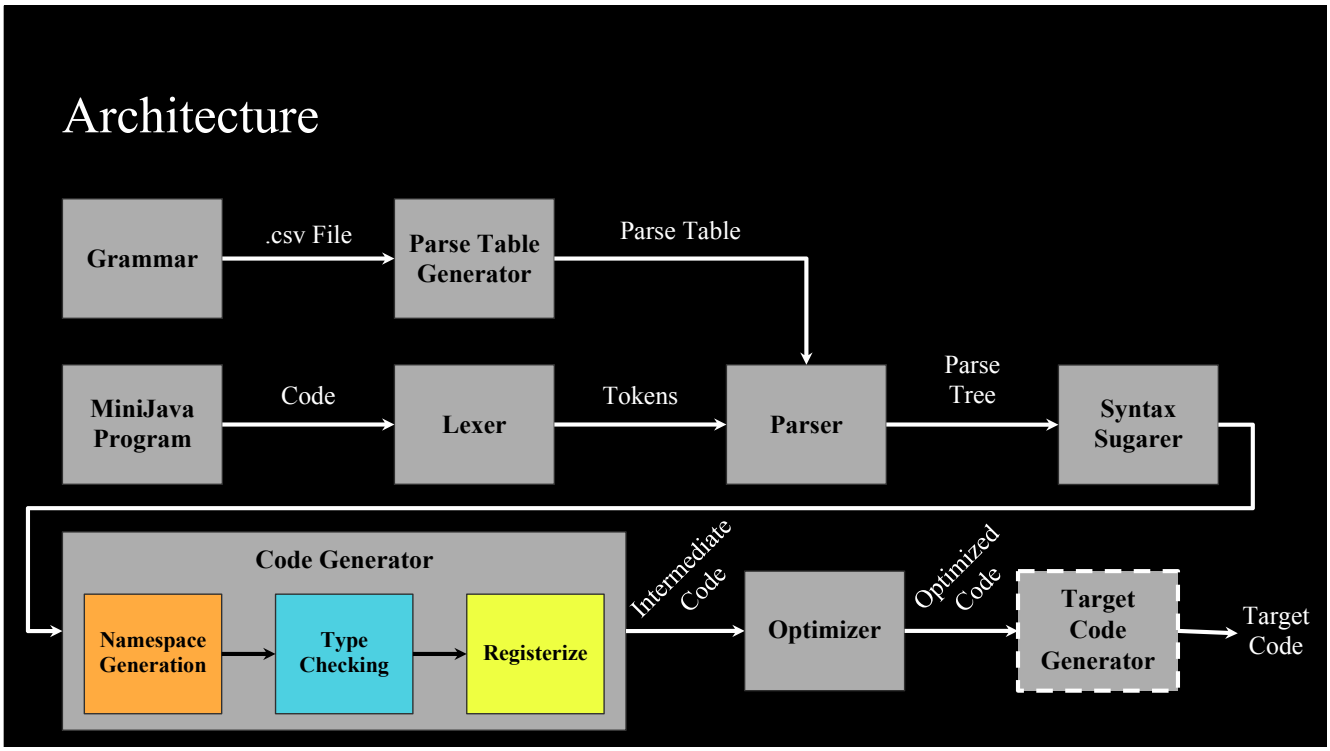
Mini-Java Compiler

Tommy McMichen and Nathan Greiner

Architecture

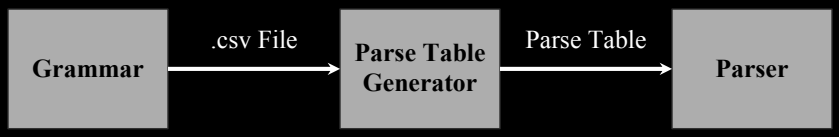


Lexer

- Artisanal, hand-coded Lexer
- Takes in MiniJava program as input
- Lexes program into ReservedWords, Integers, IDs, Delimiters, and Operators
- Outputs stream of Lexemes to Parser

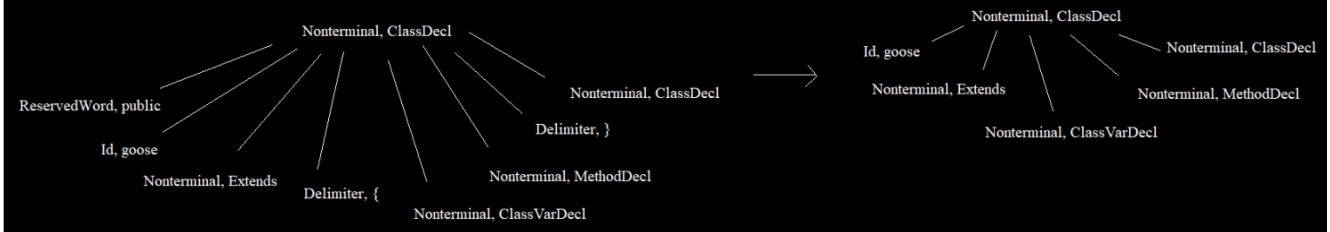
Parser

- Top-down, table-driven parser
- Parse table generated from .csv file
- Outputs parse tree to Syntax Analysis



Syntax Analysis

- Removes syntactic sugar from parse tree
- Ensures uniformity
  - i.e. every if statement consists of four children
- Outputs syntax tree to Code Generator



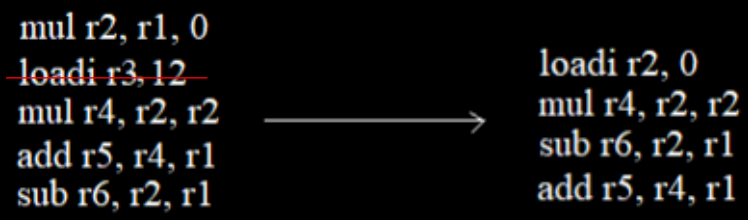
Code Generator

- Three Phase Code Generation

- RISC-Style intermediate language
  - Direct, Immediate and Inherent addressing modes
  - Designed for portability to different targets

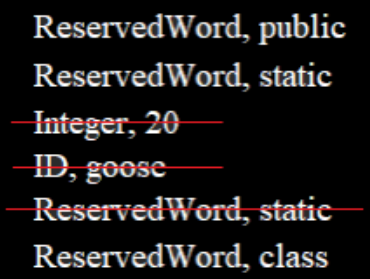
Optimization

- Simplifies Instructions
  - I.e. mult y x 2 -> add y x x
- Removes unused variables
- Schedules instructions to avoid stalls



Error Handling

- Simple “Panic mode” error handling in parser
  - Prints out error
  - Skip ahead until valid input is reached



Type Checking

- Simple pre-processing of expressions and statements
- Handled during code generation
- Casts types when allowed
  - Implements truthiness

integer == boolean → boolean

Questions?