

# Kotlin Compiler Project

November 11, 2024

## Project Overview

This project is a simple Kotlin compiler written in Haskell. It includes a lexical analyzer (scanner) implemented with the Alex library, a parser created with Happy, and the necessary logic to handle a subset of the Kotlin programming language. The compiler accepts Kotlin source code as input, tokenizes it, and parses it into an Abstract Syntax Tree (AST).

## Language Subset

The compiler currently supports a subset of Kotlin that includes the following constructs:

- **Primitive Types:** `Int`, `Long`, `Float`, `Double`, `Boolean`, `Char`, `String`
- **Operators:** Arithmetic, Increment/Decrement, Self-Operators, Comparison, Logical, Equality
- **Control Structures:** `if`, `else`, `while`
- **Functions and Variables:** `fun`, `val`, `var`, `print`, `readln`
- **Variables:** Simple variable assignments and expressions

## Compilation Process

### 1. Lexer (Scanner)

The lexer is responsible for breaking the source code into tokens, which are the building blocks for the parser. The `Lexer.x` file uses the Alex library for token generation.

## 2. Parser

The parser takes the tokens generated by the lexer and produces an Abstract Syntax Tree (AST) that represents the structure of the program. The `Parser.y` file uses the Happy library for parsing.

## 3. AST (Abstract Syntax Tree)

The AST represents the program's structure in a tree format, where each node corresponds to a construct in the language, such as an assignment, expression, or function.

## Execution

To run the compiler, use the following steps:

- **Step 1:** Write a Kotlin source code file (e.g., `input.kt`).
- **Step 2:** Run the compiler, which will tokenize the input and generate an AST.
- **Step 3:** The tokens and AST will be written to the output file (e.g., `output.txt`).

Example command:

```
$ cabal build
$ cabal run compilador.cabal
```

## Project Files

- **Lexer.x:** The lexer definition using Alex.
- **Parser.y:** The parser definition using Happy.
- **Main.hs:** The main entry point that drives the compilation process.
- **compilador.cabal:** The project configuration file for building the compiler with Cabal.

## Dependencies

The project depends on the following Haskell packages:

- `alex` – Lexer generator
- `happy` – Parser generator
- `base` – Core Haskell library

To install the dependencies, run:

```
$ cabal install alex happy
```

## Conclusion

This project demonstrates the process of building a simple compiler for a subset of Kotlin. It covers the key concepts of lexical analysis, parsing, and AST generation. Future work could expand the supported Kotlin features and improve the efficiency of the generated code.