Technical Report
on
TL 15.0 Compiler Implementation
(Project of CS5363)

1. Introduction

As part of project requirement I have implemented Compiler for TL15.0 language. In this document we provided summary of project completion status, package level functionality, sequence diagram of the implementation and features of this implementation.

2. Implementation Status

No	Module/Feature	Completion Status
1	Lexical Analyzer	**Completed**
2	Parser	**Completed**
3	Annotated AST Generation	**Completed**
4	CFG Generation	**Completed**
5	Optimization	**Completed**
6	Machine Code Generation	**Completed**
7	Testing of the Compiler	**Completed**

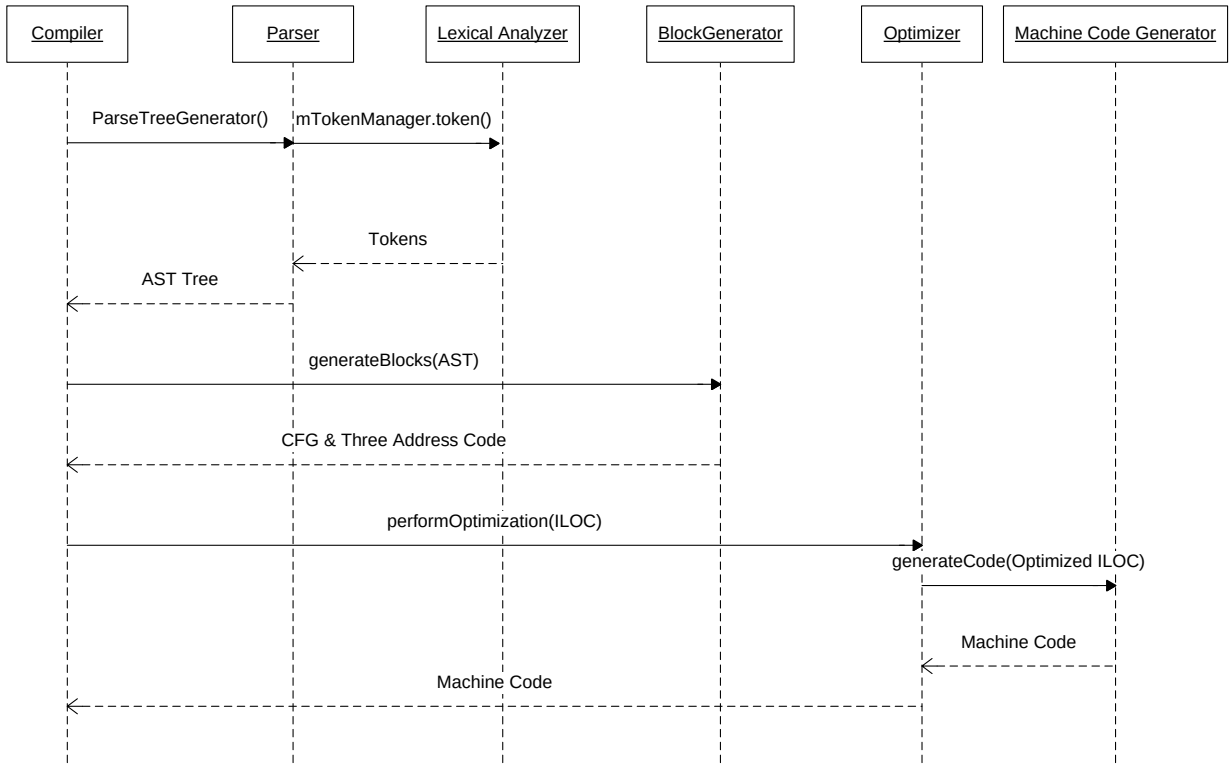
3. Implementation Overview

3.1 Package Level Functionality

In this section we have discussed about the functionalities of different implemented packages under the project implementation.

No	Package Name	Functionality
1	*com.ast*	AST Generation
2	*com.block*	Block Generation
3	*com.codegen*	Generate MIPS Code
4	*com.compiler*	Main Class to Initiate Compiler
5	*com.entities*	Entity Classes
6	*com.nodes*	Prpgram Nodes such as Declaration Node, If Node etc.
7	*com.optimization*	Perform Optimization on CFG
8	*com.parser*	Parser Functionality
9	*com.register*	Register Allocation for Variables
10	*com.scanner*	Lexical Analyzer
11	*com.utils*	Utility Classes such as File Writter, Log Printer
12	*com.visitors*	Visitor Implementation for Symantic Analysis and Print AST Tree

3.2 Sequence Diagram of the Compiler Components:



4. Features of Implementation:

a) Optimized Machine Code generated by the compiler.

b) Dead Code Elimination optimization.

c) SSH Optimization.

d) Register Uses Optimization using stack and Spelling.

e) AST Printing and Semantic checking is performed with the help of Visitor pattern that are more flexible if we want to implement further functionalities.

f) OOP based implementation of the project for future scalability.

5. GENERAL USAGE GUIDE:

1. To Run the Compiler Make Sure that JAVA is installed with proper JAVA_HOME and PATH variable value.

2. To compile and run the software run following commands in scripts folder.

For Linux Build:
./build_linux.sh
For Linux Application Execute:
./exec_linux.sh
or
./exec_linux.sh sourcefile.tl

For Windows Build:
build_win.bat
For Windows Application Execute:
exec_win.bat
or
exec_win.bat sourcefile.tl

Note: If in Linux executing .sh files faces shows permission related error, please execute following command

chmod 777 *.sh

3. You can provide TL15 source file as parameter or after running the program it will ask for source file path. Make sure that file extension contains "tl".

4. AST dot, CFG file and Execution File will be generated at TL15.0 source file directory with same name as source file; but extension will be .ast.dot, .cfg.dot and .s respectively.

5. Execution Files(*.s) can be executed in MIPS(http://pages.cs.wisc.edu/~larus/spim.html) simulator.

~END~