Amirkabir University of Technology
(Tehran Polytechnic)

Department of Computer Engineering and Information Technology

Course Project

Principles of Compiler Design

Phase two

Parser

By
Nima Davari

Professor
Dr. Mohammadreza Razzazi

Fall 2019

Overall description

Now that the Lexical Analyzer is constructed, a Parser needs to be implemented that uses the tokens returned by the scanner, in order to match the rules of the grammar. Another goal of the second phase is to make a symbol table in the same first passage through the code.

Tasks

1. Create a Yacc file.
2. In the Yacc file, define the token types as integers, and let Lex know of their values.
3. Add return values, equal to the token types, for every Regular Expression defined in the Lex file.
4. Enter the grammar rules as inputs of Yacc.
5. Set Yacc as the main entry point of the program. (Remove the main function from the Lex file.)
6. Compile the Yacc file. There should be a prompt stating the existence of 381 shift/reduce conflicts in the grammar. If this is not the case, you have done something wrong… Make sure the grammar does not contain misspellings.
7. Resolve all the conflicts. You are allowed to make changes to the grammar as long as the language remains unchanged.
8. Print out the grammar rules as they are matched by the parser, as an output for the second phase.
9. But wait! A representation of the symbol table is also required. You may use any data structures available.

The grammar can be found in a separate file.
Lex/Yacc project templates are available.
Sample symbol table data structures are available in their corresponding directories along with sample codes on how to use them.