CSE4061 – Compiler Design
Assignment #1

DUE TO: 31/03/2021 - 23:55

In this assignment, you are going to design your own language and you will be working on this language in the following assignments. Your final compiler design will be built on this language, but in your final project submission, you will submit your language description again, so if you need you will be able to make modifications on this description.

Like any programming language, your language should support the following:

- Identifiers of at least three types
- Operators: assignment, logical, unary
- Selection
- Iteration
- Function

You can check the bnf of c programming language from
http://www.cs.man.ac.uk/~pjj/bnf/c_syntax.bnf

You can work in groups of three at most. Late submissions will not be accepted. One member of your group will submit your assignment to Canvas.

Next assignments:
Assignment # 2: Lexical Analysis: 14/04/21
Assignment # 3: Syntax Analysis: 05/05/21
Assignment # 4: Semantic Analysis: 19/05/21
Assignment # 5: Intermediate Code Generation: 02/06/21

Final Project submission: 06/06/2021

Example Grammar:

program → decls compoundstmt
decls → decl **;** decls | ε
decl → **int ID** | **real ID**
stmt → ifstmt | whilestmt | assgstmt | compoundstmt
compoundstmt → **{** stmts **}**
stmts → stmt stmts | ε
ifstmt → **if (** boolexpr **) then** stmt **else** stmt
whilestmt → **while (** boolexpr **)** stmt
assgstmt → **ID =** arithexpr **;**
boolexpr → arithexpr boolop arithexpr
boolop → **<** | **>** | **<=** | **>=** | **==** | **!=**
arithexpr → arithexpr + multexpr | arithexpr − multexpr | multexpr
multexpr → multexpr * simpleexpr | multexpr / simpleexpr | simpleexpr
simpleexpr → **ID** | **INTNUM** | **REALNUM** | **(** arithexpr **)**

In this example grammar, program is the starting symbol. ID is an identifier and INTNUM and REALNUM are the variable types.

We will also need a lexical structure for our programming language. In this example, it is as follows:

- comments: Comments starts with // characters and ends with the end of line character.
- keywords: The keywords given in the grammar:
  int real if then else while
- identifiers: An identifier starts with a letter and continues with a letter or digit. A keyword cannot be an identifier, and the maximum length of an identifier is 64 characters.
- operators: The followings are operators: + - / * = == < <= > >= !=
- delimiters: The followings are delimeters: whitespace, tab, new line
- numbers: The numbers are defined as follows:
  digit ← 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
  intnumber ← digit+ (the maximum integer number is 231)
  exponent ← E (+ | - | ε) digit+ (the maximum exponent value is 128)
  fraction ← . digit+
  realnumber ← digit+ exponent | digit+ fraction (exponent | ε)