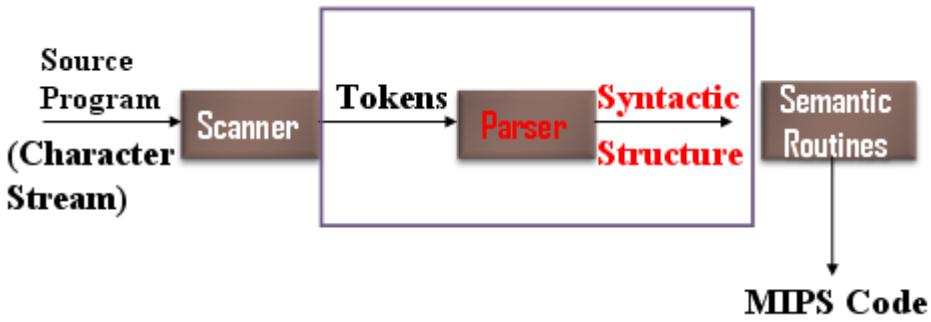
Assignment 3

The project in class is divided into three steps: **Scanner, Parser**, and **Code Generator**. Source language for the compiler project is a subset of C, and the output target code is the MIPS codes. In the second step, we list the BNF context free grammars of the language. ***Note that you are not allowed to use existing tool (eg. YACC)***, so you have to construct a parser for C entirely on your self. **C/C++ 11 is required.** Your parser must cooperate with scanner in Assignment2.



Pic1. The CFG(Control Flow Graph) of three assignments

A BNF grammar for Mini C is as follows:

1. program : var_declarations statements
2. var_declarations : var_declarations | var_declaration | / empty */*
3. var_declaration : INT declaration_list SEMI
4. declaration_list : declaration_list COMMA declaration | declaration
5. declaration : IDENTIFIER ASSIGN NUMBER | IDENTIFIER LSQUARE NUMBER RSQUARE | IDENTIFIER
6. code_block : statement | LBRACE statements RBRACE
7. statements : statements statement | statement
8. statement : assign_statement SEMI | control_statement | read_write_statement SEMI | SEMI
9. control_statement : if_statement | while_statement | do_while_statement SEMI | return_statement SEMI
10. read_write_statement : read_statement | write_statement
11. assign_statement : IDENTIFIER LSQUARE exp RSQUARE ASSIGN exp | IDENTIFIER ASSIGN exp
12. if_statement : if_stmt | if_stmt ELSE code_block

13. if_stmt : IF LPAR exp RPAR code_block
14. while_statement : WHILE LPAR exp RPAR code_block
15. do_while_statement : DO code_block WHILE LPAR exp RPAR
16. return_statement : RETURN
17. read_statement : READ LPAR IDENTIFIER RPAR
18. write_statement : WRITE LPAR exp RPAR
19. exp: INT_NUM | IDENTIFIER | IDENTIFIER LSQUARE exp LSQUARE | NOT_OP exp | exp AND_OP exp | exp OR_OP exp | exp PLUS exp | exp MINUS exp | exp MUL_OP exp | exp DIV_OP exp | exp LT exp | exp GT exp | exp EQ exp | exp NOTEQ exp | exp LTEQ exp | exp GTEQ exp | exp SHL_OP exp | exp SHR_OP exp | exp ANDAND exp | exp OROR exp | LPAR exp RPAR | MINUS exp

Review the token in scanner:

The Keywords of the language are the following:

main, if , else , break , do , while , void , int, scanf, printf

Reserve words and corresponding token
“int” {INT}
“main” { MAIN }
“void” { VOID }
“break” { BREAK }
"do" { DO }
"else" { ELSE }
"if" { IF }
"while" { WHILE }
"return" { RETRUN }
"scanf" { READ }
"printf" { WRITE }

Special symbols are the following:
& | && || <= >= == < > != = ( )
{ } . , + - * / [ ] ! >> <<

Special symbols and corresponding token

"{" { LBRACE }
"}" { RBRACE }
"[" { LSQUARE }
"]" { RSQUARE }
"(" { LPAR }
")" { RPAR }
";" { SEMI }
"+" { PLUS }

"−" { MINUS }

"*" { MUL_OP }
"/" { DIV_OP }
"&" { AND_OP }
"|" { OR_OP }
"!" { NOT_OP }
"=" { ASSIGN }
"<" { LT }
">" { GT }
"<<" { SHL_OP }
">>" { SHR_OP }
"==" { EQ }
"!=" { NOTEQ }
"<=" { LTEQ }
">=" { GTEQ }
"&&" { ANDAND }
"||" { OROR }
"," { COMMA }

Other tokens are INT_NUM, ID, defined by the following regular expressions:

digit = [0-9]

letter = [a-z A-Z]

INT_NUM = [+|-]? [digit]$^{+}$

ID = [letter]$^{+}$ [digit | letter | _]*

Evaluation

1. Technique Report 20% ; Source Code 80%

2. In source code part, If you can get Token-List from Assignment 2 (Scanner generate from LEX), you can get 60% score; If you can show the correctness of program, you can get 85% score; If you can not only show the correctness of program, but construct a parsing tree, you can get 100% score.
3. You can choose LL(1) or LALR(1), ***10 bonus points for LL(1) parser.***
4. By implementing the parser, you will **only have a** shift / reduce conflict when the grammars NO.17 【**if() / if()else**】is encountered. Therefore, you need to handle the conflict when you implement your PARSER. The default solution is
   When you match('else') , you should shift.
   When you match('other tokens') , you should reduce.
5. **Comment in Parser should be avoided.**

Format

1. **Technique Report**
   Please turn in a report**(in PDF format)** includes
   - Cover
   - Purpose
   - Method and Design
   - Experiment Environment
   - Results and Analysis
   - What you have learned

You need to describes how to execute your program in the report.

2. **Source Code**

Please upload your work to Blackboard
Create a package named as your student ID
Your package should include:
/Source code
/Makefile // If you need .
/Readme

Evaluation
Technique Report 20% (A4 not less than 10 pages)
Source Code 80%
(Base : 60% + Output Display & Source Code Comments : 20%)

Due date: 2021.04.14

Do not copy or you will get 0 point