



Trabalho de Compiladores (Etapa 2 – 4,0 pontos)

Data da entrega: 17/11/2021

Especificação da Etapa 2: Implementar um analisador sintático manualmente ou com auxílio do Yacc.

Gramática da linguagem:

<programa>	::= <lista_declarações>
<lista_declarações>	::= <lista_declarações> <declaração> <declaração>
<declaração>	::= <declaração_variáveis> <declaração_funções>
<declaração_variáveis>	::= <tipo> ID ; <tipo> ID [NUM] ;
<tipo>	::= int void
<declaração_funções>	::= <tipo> ID (<parâmetros>) <declaração_composta>
<parâmetros>	::= <lista_parâmetros> void
<lista_parâmetros>	::= <lista_parâmetros> , <param> <param>
<param>	::= <tipo> ID <tipo> ID []
<declaração_composta>	::= { <declarações_locais> <lista_comandos> }
<declarações_locais>	::= <declarações_locais> <declaração_variáveis> ε
<lista_comandos>	::= <lista_comandos> <comando> ε
<comando>	::= <declaração_expressão> <declaração_composta> <declaração_seleção> <declaração_iteração> <declaração_retorno>
<declaração_expressão>	::= <expressão> ; ;
<declaração_seleção>	::= if (<expressão>) <comando> if (<expressão>) <comando> else <comando>
<declaração_iteração>	::= while (<expressão>) <comando>
<declaração_retorno>	::= return ; return <expressão> ;
<expressão>	::= <variável> = <expressão> <expressão_simples>
<variável>	::= ID ID [<expressão>]
<expressão_simples>	::= <soma_expressão> <op_relacional> <soma_expressão> <soma_expressão>
<op_relacional>	::= <= > < > >= == !=
<soma_expressão>	::= <soma_expressão> <soma> <termo> <termo>
<soma>	::= + -
<termo>	::= <termo> <mult> <fator> <fator>
<mult>	::= * /
<fator>	::= (<expressão>) <variável> <ativação> NUM
<ativação>	::= ID (<argumentos>)
<argumentos>	::= <lista_argumentos> ε
<lista_argumentos>	::= <lista_argumentos> , <expressão> <expressão>

Observação: o analisador sintático deve funcionar em conjunto com o analisador léxico.

Entrada: programa-fonte.

Saída: árvore sintática.