Diseño de compiladores

Proyecto Nº2

Campus Santa Fe

Enrique Lira Martínez A01023351

Profesor: Dr. Víctor Manuel de la Cueva H

26 de marzo de 2019

Gramática utilizada para parser

- ➔ program -> declaration-list
- ➔ declaration-list ->declaration | declaration declaration-list
- ➔ declaration -> var-declaration | fun-declaration
- ➔ var-declaration -> type-specifier ID ; | type-specifier ID [NUM] ;
- ➔ type-specifier -> int | void
- ➔ fun-declaration -> type-specifier ID (params) compound-stmt
- ➔ param-list -> param param-list'
- ➔ param -> type-specifier ID | type-specifier ID []
- ➔ compount-stmt -> { local-declarations statement-list }
- ➔ local-declarations -> empty | var-declaration local-declarations
- ➔ statement-list -> empty | statement statement-list
- ➔ statement -> expression-stmnt | compound-stmt | selection-stmt | iteration-stmt |return stmt
- ➔ expression-stmnt -> expression ; | ;
- ➔ selection-stmt -> if (expression) statement | if (expression) statement else statement

- ➔ iteration-stmt -> while (expression) statement
- ➔ return-stmt -> return ; | return expression ;
- ➔ expression -> var = expression | simple-expression
- ➔ var -> ID | ID [expression]
- ➔ simple-expression -> additive expression relop additive-expression | additive expression

- ➔ relop -> <= | < | > | >= | == | !=
- ➔ additive-expression -> term additive-expression’
- ➔ additive-expression’ -> addop term additive-expression’ | empty
- ➔ addop -> + | -
- ➔ term -> factor | term’
- ➔ term’ -> empty | mulop factor term’
- ➔ mulop *| /
- ➔ factor -> (expression) | var | call | NUM
- ➔ call -> ID (args)
- ➔ args -> arg-list | empty
- ➔ arg-list -> expression arg-list'