

Tecnológico de Monterrey

Campus Toluca

A01366501 | César Iván Pedrero Martínez

Compiladores

Proyecto 2:

Parser: Gramática EBNF

# EBNF

* program -> declaration-list
* declaration-list -> declaration { declaration }
* declaration -> var-declaration | fun-declaration
* var-declaration -> type-specifier ID [ “[“ NUM “]” ]
* fun-declaration -> type-specifier ID “(“ params “)” compound-stmt
* type-specifier -> int|void
* params -> param-list | void
* param-list -> param { , param }
* param -> type-specifier ID [ “[“ “]” ]
* compound-stmt -> “{“ local-declarations statement-list “}”
* local-declarations -> [ var-declarations ]
* statement-list -> [ statement-list ]
* statement -> expression-stmt | compound-stmt | selection-stmt | iteration-stmt | return-stmt
* expression-stmt -> [ expression ] ;
* compound-stmt -> “{“ local-declarations statement-list “}”
* selection-stmt -> if ( expression ) statement [ else statement ]
* iteration-stmt -> while ( expression ) statement
* return-stmt -> return [ expression ] ;
* expression -> var = expression | simple-expression
* var -> ID [ “[“ expression “]” ]
* simple-expression -> additive-expression [ relop additive expression ]
* relop -> <= | < | > | >= | == | !=
* additive-expression -> [ additive-expression addop ] term
* addop -> + | -
* term -> [ term mulop ] factor
* mulop -> \* | /
* factor -> ( expression ) | var | call | NUM
* call -> ID ( args )
* args -> [ arg-list ]
* arg-list -> expression [ , expression ]