Centro Universitário de Belo Horizonte – Uni-BH
Curso: Ciência da Computação
UC: Teoria da Computação e Compiladores
Professor: Gustavo Alves Fernandes

Gramática Corrigida da linguagem PasC
(com enumeração das regras)

prog → “program” “id” body 1
body → decl-list “{“ stmt-list “}” 2
decl-list → decl “;” decl-list 3 | ε 4
decl → type id-list 5
type → “num” 6 | “char” 7
id-list → “id” id-list’ 8
id-list’ → “,” id-list 9 | ε 10

stmt-list → stmt “;” stmt-list 11 | ε 12
stmt → assign-stmt 13 | if-stmt 14 | while-stmt 15 | read-stmt 16 | write-stmt 17
assign-stmt → “id” “=” simple_expr 18
if-stmt → “if” “(“ expression “)” “{“ stmt-list “}” if-stmt’ 19
if-stmt’ → “else” “{“ stmt-list “}” 20 | ε 21
while-stmt → stmt-prefix “{“ stmt-list “}” 22
stmt-prefix → “while” “(“ expression “)” 23
read-stmt → “read” “id” 24
write-stmt → “write” simple-expr 25

expression → simple-expr expression’ 26
expression’ → logop simple-expr expression’ 27 | ε 28
simple-expr → term simple-exp’29
simple-exp’ → relop term simple-exp’ 30 | ε 31
term → factor-b term’ 32
term’ → addop factor-b term’ 33 | ε 34
factor-b → factor-a factor-b’ 35
factor-b’ → mulop factor-a factor-b’ 36 | ε 37
factor-a → factor 38 | not factor 39
factor → “id” 40 | constant 41 | “(“ expression “)” 42

logop → “or” 43 | “and” 44
relop → “==”45 | “>” 46 | “>=” 47 | “<” 48 | “<=” 49 | “!=” 50
addop → “+” 51 | “-” 52
mulop → “*” 53 | “/” 54
constant → “num_const” 55 | “char_const” 56

Cáculo FIRST e FOLLOW

	FIRST	FOLLOW
prog	“program”	“$”
body	“num”, “char”, “{“	“$”
decl-list	“num”, “char”, “ε”	“{“
decl	“num”, “char”	“;”
type	“num”, “char”	“id”
id-list	“id”	“;”
id-list’	“,”, “ε”	“;”
stmt-list	“id”, “if”, “while”, “read”, “write”, “ε”	“}”
stmt	“id”, “if”, “while”, “read”, “write”	“;”
assign-stmt	“id”	“;”
if-stmt	“if”	“;”
if-stmt’	“else”, “ε”	“;”
while-stmt	“while”	“;”
stmt-prefix	“while”	“{“
read-stmt	“read”	“;”
write-stmt	“write”	“;”
expression	“id”, “num_const”, “char_const”, “(“, “not”	“)”
expression’	“or”, “and”, “ε”	“)”
simple-expr	“id”, “num_const”, “char_const”, “(“, “not”	“or”, “and”, “)”
simple-expr’	“==”, “>”, “>=”, “<”, “<=”, “!=”, “ε”	“or”, “and”, “)”
term	“id”, “num_const”, “char_const”, “(“, “not”	“==”, “>”, “>=”, “<”, “<=”, “!=”, “or”, “and”, “)”
term’	“+”, “-”, “ε”	“==”, “>”, “>=”, “<”, “<=”, “!=”, “or”, “and”, “)”
factor-b	“id”, “num_const”, “char_const”, “(“, “not”	“+”, “-”, “==”, “>”, “>=”, “<”, “<=”, “!=”, “or”, “and”, “)”
factor-b’	“*”, “/”, “ε”	“+”, “-”, “==”, “>”, “>=”, “<”, “<=”, “!=”, “or”, “and”, “)”
factor-a	“id”, “num_const”, “char_const”, “(“, “not”	“*”, “/”, “+”, “-”, “==”, “>”, “>=”, “<”, “<=”, “!=”, “or”, “and”, “)”
factor	“id”, “num_const”, “char_const”, “(“	“*”, “/”, “+”, “-”, “==”, “>”, “>=”, “<”, “<=”, “!=”, “or”, “and”, “)”

logop	“or”, “and”	“id”, “num_const”, “char_const”, “(“, “not”
relop	“==”, “>”, “>=”, “<”, “<=”, “!=”	“id”, “num_const”, “char_const”, “(“, “not”
addop	“+”, “-”, “or”	“id”, “num_const”, “char_const”, “(“, “not”
mulop	“*”, “/”, “and”	“id”, “num_const”, “char_const”, “(“, “not”
constant	“num_const”, “char_const”	“*”, “/”, “+”, “-”, “==”, “>”, “>=”, “<”, “<=”, “!=”, “or”, “and”, “)”