

Gramática Avance 1

Expresiones regulares

GRAMÁTICA para un PROGRAMA

PROGRAMA -> program id ; VARS FUNC main BODY end

Gramática para un VARS

VARS -> VAR TYPE TIPO VARS2 id ;

VARS2 -> id ; | id , VARS2

GRAMÁTICA para un TIPO

TIPO -> int | float | char

GRAMÁTICA para un BLOQUE

BODY -> ESTATUTO BODY | ESTATUTO

GRAMÁTICA para un ESTATUTO

ESTATUTO -> ASIGNACIÓN | LLAMADA | CONDICIÓN | RETORNO | LECTURA |

CONDICION | CICLO_W | CICLO_F | FUNC_ESP

GRAMÁTICA para una ASIGNACIÓN

ASIGNACIÓN -> VARIABLE = EXP ;

GRAMÁTICA para una LLAMADA:

LLAMADA -> id (EXP LLAMADA2)

LLAMADA2 -> , EXP || eps

GRAMÁTICA para un RETORNO

RETORNO -> return (EXP) ;

GRAMÁTICA para una LECTURA:

LECTURA -> read (VARIABLE) ;

GRAMÁTICA para una ESCRITURA:

ESCRITURA -> write (EXP ESCRITURA2) ;

ESCRITURA2 -> letrero | , ESCRITURA2 | eps

GRAMÁTICA para una CONDICIÓN:

CONDICIÓN -> if (EXP) then { ESTATUTO ; ESTATUTO2 } | if (EXP) then { ESTATUTO ; } else { ESTATUTO ESTATUTO2 }

Rodolfo Ramírez Cedillo

A00820906

ESTATUTO2 -> ESTATUTO ; ESTATUTO2 | eps

GRAMÁTICA para un CICLO_F:

CICLO_F -> for ASIGNACION to EXP do { ESTATUTO ; ESTATUTO2 }

GRAMÁTICA para un CICLO_W:

CICLO_W -> while (EXP) do { ESTATUTO ; ESTATUTO2 }

GRAMÁTICA para un EXP:

EXP -> T_EXP EXP2

EXP2 -> or T_EXP | eps

GRAMÁTICA para un T_EXP:

T_EXP -> G_EXP T_EXP2

T_EXP2 -> && G_EXP | eps

GRAMÁTICA para un G_EXP:

G_EXP -> M_EXP M_EXP2

M_EXP2 -> > M_EXP | < M_EXP | == M_EXP | != M_EXP | eps

GRAMÁTICA para un F:

F -> (EXP) | CTE_I | CTE_F | CTE_CHAR | VARIABLE | LLAMADA