

Análisis LL1 de Gramatica.II1

Analizando símbolo A
Analizando producción A -> T id K
Analizando símbolo T
Analizando producción T -> number
FIRST de T -> number = { number }
Analizando producción T -> boolean
FIRST de T -> boolean = { boolean }
Analizando producción T -> string
FIRST de T -> string = { string }
FIRST de T = { boolean number string }
FIRST de A -> T id K = { boolean number string }
Analizando producción A -> lambda
FIRST de A -> lambda = { lambda }
FIRST de A = { boolean number string lambda }
Calculando FOLLOW de A
FOLLOW de A = { } }
Analizando símbolo B
Analizando producción B -> let T id ;
FIRST de B -> let T id ; = { let }
Analizando producción B -> if (R) S
FIRST de B -> if (R) S = { if }
Analizando producción B -> while (R) { C }
FIRST de B -> while (R) { C } = { while }
Analizando producción B -> S
Analizando símbolo S
Analizando producción S -> id Sa
FIRST de S -> id Sa = { id }
Analizando producción S -> return X ;
FIRST de S -> return X ; = { return }
Analizando producción S -> alert (R) ;
FIRST de S -> alert (R) ; = { alert }
Analizando producción S -> input (id) ;
FIRST de S -> input (id) ; = { input }
FIRST de S = { alert id input return }
FIRST de B -> S = { alert id input return }
FIRST de B = { alert id if input let return while }
Analizando símbolo C
Analizando producción C -> B C
FIRST de C -> B C = { alert id if input let return while }
Analizando producción C -> lambda
FIRST de C -> lambda = { lambda }
FIRST de C = { alert id if input let return while lambda }
Calculando FOLLOW de C
FOLLOW de C = { } }
Analizando símbolo Ep
Analizando producción Ep -> != T Ep
FIRST de Ep -> != T Ep = { != }
Analizando producción Ep -> lambda
FIRST de Ep -> lambda = { lambda }
FIRST de Ep = { != lambda }
Calculando FOLLOW de Ep
FOLLOW de Ep = { }
Analizando símbolo F
Analizando producción F -> function H id (A) { C }
FIRST de F -> function H id (A) { C } = { function }
FIRST de F = { function }
Analizando símbolo H
Analizando producción H -> T
FIRST de H -> T = { boolean number string }
Analizando producción H -> lambda
FIRST de H -> lambda = { lambda }
FIRST de H = { boolean number string lambda }
Calculando FOLLOW de H
FOLLOW de H = { id }
Analizando símbolo K
Analizando producción K -> , T id K
FIRST de K -> , T id K = { , }
Analizando producción K -> lambda
FIRST de K -> lambda = { lambda }
FIRST de K = { , lambda }
Calculando FOLLOW de K
FOLLOW de K = { } }
Analizando símbolo L
Analizando producción L -> R Q
Analizando símbolo R
Analizando producción R -> U Ra
Analizando símbolo U
Analizando producción U -> V Ua
Analizando símbolo V
Analizando producción V -> id Va
FIRST de V -> id Va = { id }
Analizando producción V -> entero
FIRST de V -> entero = { entero }
Analizando producción V -> cadena
FIRST de V -> cadena = { cadena }
Analizando producción V -> true
FIRST de V -> true = { true }

Analizando producción V -> false
 FIRST de V -> false = { false }
 Analizando producción V -> (R)
 FIRST de V -> (R) = { (}
 Analizando producción V -> ! Vb
 FIRST de V -> ! Vb = { ! }
 FIRST de V = { ! (cadena entero false id true }
 FIRST de U -> V Ua = { ! (cadena entero false id true }
 FIRST de U = { ! (cadena entero false id true }
 FIRST de R -> U Ra = { ! (cadena entero false id true }
 FIRST de R = { ! (cadena entero false id true }
 FIRST de L -> R Q = { ! (cadena entero false id true }
 Analizando producción L -> lambda
 FIRST de L -> lambda = { lambda }
 FIRST de L = { ! (cadena entero false id true lambda }
 Calculando FOLLOW de L
 FOLLOW de L = { } }
 Analizando símbolo P
 Analizando producción P -> B P
 FIRST de P -> B P = { alert id if input let return while }
 Analizando producción P -> F P
 FIRST de P -> F P = { function }
 Analizando producción P -> eof
 FIRST de P -> eof = { eof }
 FIRST de P = { alert eof function id if input let return while }
 Analizando símbolo Q
 Analizando producción Q -> , R Q
 FIRST de Q -> , R Q = { , }
 Analizando producción Q -> lambda
 FIRST de Q -> lambda = { lambda }
 FIRST de Q = { , lambda }
 Calculando FOLLOW de Q
 FOLLOW de Q = { } }
 Analizando símbolo Ra
 Analizando producción Ra -> = R
 FIRST de Ra -> = R = { = }
 Analizando producción Ra -> != R
 FIRST de Ra -> != R = { != }
 Analizando producción Ra -> lambda
 FIRST de Ra -> lambda = { lambda }
 FIRST de Ra = { != lambda }
 Calculando FOLLOW de Ra
 Calculando FOLLOW de R
 Calculando FOLLOW de X
 FOLLOW de X = { ; }
 FOLLOW de R = { } , ; }
 FOLLOW de Ra = { } , ; }
 Analizando símbolo Sa
 Analizando producción Sa -> = R ;
 FIRST de Sa -> = R ; = { = }
 Analizando producción Sa -> != R ;
 FIRST de Sa -> != R ; = { != }
 Analizando producción Sa -> (L) ;
 FIRST de Sa -> (L) ; = { (}
 Analizando producción Sa -> /= R ;
 FIRST de Sa -> /= R ; = { /= }
 FIRST de Sa = { != (/= = }
 Analizando símbolo Ua
 Analizando producción Ua -> + U
 FIRST de Ua -> + U = { + }
 Analizando producción Ua -> lambda
 FIRST de Ua -> lambda = { lambda }
 FIRST de Ua = { + lambda }
 Calculando FOLLOW de Ua
 Calculando FOLLOW de U
 FOLLOW de U = { !=) , ; = }
 FOLLOW de Ua = { !=) , ; = }
 Analizando símbolo Va
 Analizando producción Va -> (L)
 FIRST de Va -> (L) = { (}
 Analizando producción Va -> lambda
 FIRST de Va -> lambda = { lambda }
 FIRST de Va = { (lambda }
 Calculando FOLLOW de Va
 Calculando FOLLOW de V
 FOLLOW de V = { !=) + , ; = }
 FOLLOW de Va = { !=) + , ; = }
 Analizando símbolo Vb
 Analizando producción Vb -> true
 FIRST de Vb -> true = { true }
 Analizando producción Vb -> false
 FIRST de Vb -> false = { false }
 Analizando producción Vb -> id
 FIRST de Vb -> id = { id }
 Analizando producción Vb -> (R)
 FIRST de Vb -> (R) = { (}
 FIRST de Vb = { (false id true }
 Analizando símbolo X
 Analizando producción X -> R
 FIRST de X -> R = { ! (cadena entero false id true }
 Analizando producción X -> lambda
 FIRST de X -> lambda = { lambda }

FIRST de $X = \{ ! (\text{cadena entero false id true lambda}) \}$

Análisis concluido satisfactoriamente