

Análisis LL1 de gramatica.ll1

Analizando símbolo A
Analizando producción A -> (A1)
FIRST de A -> (A1) = { (}
Analizando producción A -> lambda
FIRST de A -> lambda = { lambda }
FIRST de A = { (lambda }
Calculando FOLLOW de A
Calculando FOLLOW de O
Analizando símbolo R5
Analizando producción R5 -> J O R5
Analizando símbolo J
Analizando producción J -> +
FIRST de J -> + = { + }
Analizando producción J -> -
FIRST de J -> - = { - }
FIRST de J = { + - }
FIRST de R5 -> J O R5 = { + - }
Analizando producción R5 -> lambda
FIRST de R5 -> lambda = { lambda }
FIRST de R5 = { + - lambda }
Calculando FOLLOW de R5
Calculando FOLLOW de R4
Analizando símbolo R3
Analizando producción R3 -> == R4 R3
FIRST de R3 -> == R4 R3 = { == }
Analizando producción R3 -> lambda
FIRST de R3 -> lambda = { lambda }
FIRST de R3 = { == lambda }
Calculando FOLLOW de R3
Calculando FOLLOW de R2
Analizando símbolo R1
Analizando producción R1 -> && R2 R1
FIRST de R1 -> && R2 R1 = { && }
Analizando producción R1 -> lambda
FIRST de R1 -> lambda = { lambda }
FIRST de R1 = { && lambda }
Calculando FOLLOW de R1
Calculando FOLLOW de R
Analizando símbolo A2
Analizando producción A2 -> , R A2
FIRST de A2 -> , R A2 = { , }
Analizando producción A2 -> lambda
FIRST de A2 -> lambda = { lambda }
FIRST de A2 = { , lambda }
Calculando FOLLOW de A2
Calculando FOLLOW de A1
FOLLOW de A1 = { }
FOLLOW de A2 = { }
Calculando FOLLOW de U2
FOLLOW de U2 = { ; }
Calculando FOLLOW de F1
FOLLOW de F1 = { ; }
Calculando FOLLOW de F2
FOLLOW de F2 = { }
FOLLOW de R = { } , ; }
FOLLOW de R1 = { } , ; }
FOLLOW de R2 = { &&) , ; }
FOLLOW de R3 = { &&) , ; }
FOLLOW de R4 = { &&) , ; == }
FOLLOW de R5 = { &&) , ; == }
FOLLOW de O = { &&) + , - ; == }
FOLLOW de A = { &&) + , - ; == }
Analizando símbolo A1
Analizando producción A1 -> R A2
Analizando símbolo R
Analizando producción R -> R2 R1
Analizando símbolo R2
Analizando producción R2 -> R4 R3
Analizando símbolo R4
Analizando producción R4 -> O R5
Analizando símbolo O
Analizando producción O -> (R)
FIRST de O -> (R) = { (}
Analizando producción O -> id A
FIRST de O -> id A = { id }
Analizando producción O -> numero
FIRST de O -> numero = { numero }
Analizando producción O -> cadena
FIRST de O -> cadena = { cadena }
FIRST de O = { (cadena id numero }
FIRST de R4 -> O R5 = { (cadena id numero }
FIRST de R4 = { (cadena id numero }
FIRST de R2 -> R4 R3 = { (cadena id numero }
FIRST de R2 = { (cadena id numero }
FIRST de R -> R2 R1 = { (cadena id numero }
FIRST de R = { (cadena id numero }
FIRST de A1 -> R A2 = { (cadena id numero }

Analizando producción A1 -> lambda
FIRST de A1 -> lambda = { lambda }
FIRST de A1 = { (cadena id numero lambda }
Analizando símbolo B
Analizando producción B -> var T id C
FIRST de B -> var T id C = { var }
Analizando producción B -> if (R) U
FIRST de B -> if (R) U = { if }
Analizando producción B -> U
Analizando símbolo U
Analizando producción U -> input id ;
FIRST de U -> input id ; = { input }
Analizando producción U -> output R ;
FIRST de U -> output R ; = { output }
Analizando producción U -> return U2 ;
FIRST de U -> return U2 ; = { return }
Analizando producción U -> id U1
FIRST de U -> id U1 = { id }
FIRST de U = { id input output return }
FIRST de B -> U = { id input output return }
Analizando producción B -> for (F1 ; R ; F2) { Q }
FIRST de B -> for (F1 ; R ; F2) { Q } = { for }
FIRST de B = { for id if input output return var }
Analizando símbolo C
Analizando producción C -> ;
FIRST de C -> ; = { ; }
Analizando producción C -> L R ;
Analizando símbolo L
Analizando producción L -> +=
FIRST de L -> += = { += }
Analizando producción L -> =
FIRST de L -> = = { = }
FIRST de L = { += = }
FIRST de C -> L R ; = { += = }
FIRST de C = { += ; = }
Analizando símbolo D
Analizando producción D -> function H id (D1) { Q }
FIRST de D -> function H id (D1) { Q } = { function }
FIRST de D = { function }
Analizando símbolo D1
Analizando producción D1 -> D2
Analizando símbolo D2
Analizando producción D2 -> D3 D4
Analizando símbolo D3
Analizando producción D3 -> T id
Analizando símbolo T
Analizando producción T -> boolean
FIRST de T -> boolean = { boolean }
Analizando producción T -> int
FIRST de T -> int = { int }
Analizando producción T -> string
FIRST de T -> string = { string }
FIRST de T = { boolean int string }
FIRST de D3 -> T id = { boolean int string }
FIRST de D3 = { boolean int string }
FIRST de D2 -> D3 D4 = { boolean int string }
FIRST de D2 = { boolean int string }
FIRST de D1 -> D2 = { boolean int string }
Analizando producción D1 -> void
FIRST de D1 -> void = { void }
FIRST de D1 = { boolean int string void }
Analizando símbolo D4
Analizando producción D4 -> , D2
FIRST de D4 -> , D2 = { , }
Analizando producción D4 -> lambda
FIRST de D4 -> lambda = { lambda }
FIRST de D4 = { , lambda }
Calculando FOLLOW de D4
Calculando FOLLOW de D2
Calculando FOLLOW de D1
FOLLOW de D1 = { }
FOLLOW de D2 = { }
FOLLOW de D4 = { }
Analizando símbolo F1
Analizando producción F1 -> id L R
FIRST de F1 -> id L R = { id }
Analizando producción F1 -> lambda
FIRST de F1 -> lambda = { lambda }
FIRST de F1 = { id lambda }
Analizando símbolo F2
Analizando producción F2 -> id L R
FIRST de F2 -> id L R = { id }
Analizando producción F2 -> lambda
FIRST de F2 -> lambda = { lambda }
FIRST de F2 = { id lambda }
Analizando símbolo H
Analizando producción H -> T
FIRST de H -> T = { boolean int string }
Analizando producción H -> void
FIRST de H -> void = { void }
FIRST de H = { boolean int string void }
Analizando símbolo Q

Analizando producción Q -> B Q
FIRST de Q -> B Q = { for id if input output return var }
Analizando producción Q -> lambda
FIRST de Q -> lambda = { lambda }
FIRST de Q = { for id if input output return var lambda }
Calculando FOLLOW de Q
FOLLOW de Q = { }
Analizando símbolo S
Analizando producción S -> B S
FIRST de S -> B S = { for id if input output return var }
Analizando producción S -> D S
FIRST de S -> D S = { function }
Analizando producción S -> lambda
FIRST de S -> lambda = { lambda }
FIRST de S = { for function id if input output return var lambda }
Calculando FOLLOW de S
FOLLOW de S = { \$ (final de cadena) }
Analizando símbolo U1
Analizando producción U1 -> L R ;
FIRST de U1 -> L R ; = { += = }
Analizando producción U1 -> (A1) ;
FIRST de U1 -> (A1) ; = { (}
FIRST de U1 = { (+= = }
Analizando símbolo U2
Analizando producción U2 -> R
FIRST de U2 -> R = { (cadena id numero }
Analizando producción U2 -> lambda
FIRST de U2 -> lambda = { lambda }
FIRST de U2 = { (cadena id numero lambda }

Análisis concluido satisfactoriamente