Compiladores 2020

Venegas Guerrero Fatima Alejandra

Definición Dirigida por sintaxis

PRODUCCION	REGLAS SEMANTICAS
P → D F	dir = 0 Temporales = 0

D → T L D	Si !existe(T/id) entonces
	  insertar(T/id)
	Sino
	  error(T/id duplicado)
	Fin si

D → ε	

D → TR LV	Si !existe(TR) entonces
	  insertar(TR)
	Sino
	  error(TR duplicado)
	Fin si

TA →[num]TA	TA.tipo := array(num val,TA1,tipo); TA.tipo := num val x TA1.ancho)
TA → ε	
TL →id TL’	TL.matriz := id.lugar; TL.lugar := id.lugar; TL.ndim := 1
LV →id LV	LV.matriz := id.lugar; LV.lugar := LV.lugar; LV.ndim := 1
LV → ε	
BA → ent	BA.tipo := ent;
BA → real	BA.tipo := real;
BA → dreal	BA.tipo := dreal;
BA → car	BA.tipo := car;
BA → sin	BA.tipo := sin;

F → def TI id () {Q}	generar_etiqueta(TI) generar_etiqueta(inicio) generar_etiqueta(fin) generar_etiqueta(Q.lnext,fin)
F → ε	
S → if(B) {Q}	retroceso(B.ltrue, Q.first) S.lnext = combinar(B.lfalse, Q.lnext) S.first = B.first
S → if(B) {Q1} else G {Q2}	retroceso(B.ltrue, Q_1.first) retroceso(B.lfalse, Q_2.first) temp = combinar(Q_1.lnext, G.next) S.lnext = combinar(temp, Q_2.lnext)
S → while (B) Q	retroceso(Q.ltrue, B.first) retroceso(B.ltrue, Q.first) S.next = B.lfalse genera codigo('goto' B.firs)
S → S1 S2	S_1.next = newLabel() S_2.next = S.next S.codigo = S_1.codigo ‖ label(S_1.next) ‖ S_2.codigo
S → ID = E :	si existe(id) entonces inst =generar codigo(id '=' E.dir) si E.first != -1 entonces S.first = E.first sino S.first = inst fin si sino error("El id no ha sido declarado") fin si

E → E1 + E2	E.dir=newTemp() inst = genera codig(E.dir '=' E_1.dir '+' E_2.dir) si E_1.first != -1 entonces E.first = E_1.first sino si E_2.first != -1 entonces E.first = E_2.first sino E.first = inst fin si fin si
E → E1 - E2	E.dir=newTemp() inst = genera codig(E.dir '=' E_1.dir '-' E_2.dir) si E_1.first != -1 entonces E.first = E_1.first sino si E_2.first != -1 entonces E.first = E_2.first sino E.first = inst fin si fin si
E → E1 * E2	E.dir = new Temp() inst = genera codig(E.dir '=' E_1.dir '*' E_2.dir) si E_1.first != -1 entonces E.first = E_1.first sino si E_2.first != -1 entonces E.first = E_2.first sino E.first = inst fin si fin si

E → E1 / E2	E.dir = new Temp() inst = genera codig(E.dir '=' E_1.dir '/' E_2.dir) si E_1.first != -1 entonces E.first = E_1.first sino si E_2.first != -1 entonces E.first = E_2.first sino E.first = inst fin si fin si
E → E1 % E2	E.dir = new Temp() inst = genera codig(E.dir '=' E_1.dir '%' E_2.dir) si E_1.first != -1 entonces E.first = E_1.first sino si E_2.first != -1 entonces E.first = E_2.first sino E.first = inst fin si fin si
E → num	E.tipo := num;
E → (E1)	E1.verdadera := E.verdadera E1.false := E.verdadera E.codigo := E1.codigo
E → variable	E.tipo := variable;
E → cadena	E.tipo := cadena;
E → caracter	E.tipo := caracter;
E → ε	
B →or B1	B1.verdadera : = B.verdadera; B1.falsa := etiquetanueva; B.verdadera := E.verdadera; B.falsa := B.falsa B.codigo := B1.codigo

B →and B1	B1.verdadera : = etiqueta nueva; B1.falsa := B1.falsa; B.verdadera := B;verdadero B.falsa := B.falsa B.codigo := B1.codigo
U →verdadero	U.codigo := gen(‘goto’ U.verdadero)
U →falso	U.codigo := gen(‘goto’ U.falso)

Se realizo tomando la explicación del libro “Compiladroes : principios ,técnicas y herramientas,Alfred V Aho”