Universidad Nacional Autónoma de México

Facultad de Ingeniería

Compiladores

Grupo: 01

Definición Dirigida por Sintaxis

Tapia Álvarez Jorge Saúl

Amalfi Figueroa Issac

Alvarez Solís Iván

Escamilla Jaime Neftalí

Fecha de Entrega: 7 de junio de 2020

Profesor: ING. Mercado Martínez Adrián Ulises

Producción	Reglas Semánticas
P → D F	STS.push(nuevaTS()) STT.push(nuevaTT()) Dir=0 P.code=s.code TablaDeCadenas= new TablaCadenas
D → TL_v;	Tipo=t.Tipo
T → T_B C	Base= $T_B.base$ T.tipo=C.tipo
T → struct {D}	STS.push(nuevaTS()) STT.push(nuevaTT()) Sdir.push(dir) dir=0 dir = Sdir.pop() Ts = STS.pop() Tt = STT.pop() ts.tt = Tt T.ipo = STT.getCima().append('struct', tam, Ts)
.T_B → int	.T_B.base =int
.T_B → float	.T_B. base =float
.T_B → dreal	.T_B.base=dreal
.T_B → car	.T_B.base=car
.T_B → sin	.T_B.=sin
C → [num] C_1	**Si** num.tipo = entero **Entonces** **Si** num.dir > 0 **Entonces** C.tipo = TT.append("array", num.dir, C_1 .tipo) **Sino** Error("…") **Fin Si** **Sino** Error("…") **Fin** Si
C → ε	C.tipo =Base
L_{V1}→ L_{V1}, id	**Si** !Ts.existe(id) **Entonces** STS.getCima().append(id, dir, Tipo, 'var',nulo, -1) Dir ← dir + STT.getCima().getTam(Tipo) **Sino** Error(…) **Fin Si**
L_V → id	**Si** !Ts.existe(id) **Entonces** STS.getCima().append(id, dir, Tipo, 'var',nulo, -1) Dir ← dir + STT.getCima().getTam(Tipo) **Sino** Error(…) **Fin Si**
F → define T id (A) { D S }	**Si** no STS.getCima().existe(id) Entonces STS.push(nuevaTS())

	Sdir.push(dir) dir=0 lista_retorno = nuevaLista() **Si** cmpRet(lista_retorno, T.tipo) **Entonces** L =nuevaEtiqueta() backpatch(S.nextlist, L) F.code = etiqueta(id) || S.code || etiqueta(L) **Sino** Error("el valor no corresponde al tipo de la Función") **Fin Si** STS.pop() dir= Sdir.pop() **Sino** Error("El id ya fue declarado") **Fin Si**
A → ε	A.lista=nulo A.num=0
A → L_a	A.lista= $L_a.lista$ A.num= L_a.num
L_a → L_{a1}, T id	L_a.lista = L_a.lista L_a . $lista.append$(T.tipo) L_a.num =L_{a1}.num+1
L_a → T id	L_a.lista = L_a.lista L_a . $lista.append$(T.tipo) L_a.num =L_{a1}.num+1
T_a→ base param_arr	T.base= base.tipo T_a.tipo =param_arr.tipo
Param_arr → [] param arr	Parram arr.tipo = StackTT.getCima().addTipo("array", -, param arr1.tipo)
Param_arr → ε	Param_arr = nulo Param_arr.tipo =base
S → S_1S_2	L =nuevaEtiqueta() backpatch(S_1.nextlist, L) S.nextlist = S_2 .nextlist S.code =S_1 .code ||etiqueta(L) || S_2 .code
S → **if**(B) S_1	L =nuevaEtiqueta() backpatch(B.truelist,) S.nextlist= combinar(B.falselist,S_1 .nextlist) S.code = B.code || etiqueta(L) || S_1 .code
S → **if** (B) S_1**else** S_2	L_1= nuevaEtiqueta() L_2 = nuevaEtiqueta() backpatch(B.truelist, L_1) backpatch(B.falselist, L_2) S.nextlist = combinar(S_1 .nextlist, S_2 .nextlist) S.code = B.code || etiqueta(L_1) || S_1.code || gen('goto' S_1.nextlist[0]) || etiqueta(L_2) || S_2 .code

S → while (B) S_1	L_1 = nuevaEtiqueta() L_2= nuevaEtiqueta() backpatch(S_1 .nextlist, L_1) backpatch(B.truelist, L_2) S.nextlist = B.falselist S.code = etiqueta(L_1) ||B.code || etiqueta(L_2) || S_1.code || gen('goto' S_1.nextlist[0]) || S_2 .code
S → {S_1}	S.nextlist =nulo **Si** TS.existe(id) **Entonces** tipo_id = TS.getTipo(id) t = reducir(E.dir, E.tipo, tipo_id) gen(id '=' t) **Sino** error("El id no ha sido declarado") **Fin Si**
S → L_a=E;	S.nextlist = nulo t= reducir(E.dir, E.tipo , L_a .tipo) gen(L_a .base'[L_a.dir']' '=' t)
S → return E;	S.nextlist = nulo lista_retorno.append(E.tipo) S.code = gen(return E.dir)
B → B_1 || B_2	L = nuevaEtiqueta() backpatch(B_1.falselist, L) B.truelist = combinar(B_1.truelist, B_2.truelist) B.falselist = B_2.falselist B.code = B_1.code || etiqueta(L) || B_2.code
B → B_1 && B_2	L = nuevaEtiqueta() backpatch(B_1.truelist, L) B.truelist = B_2 .truelist B.falselist = combinar(B_1 .falselist, B_2 .falselist) B.code = B_1.code || etiqueta(L) || B_2 .code
B → ! B_1	B.truelist = B_1 .falselist B.falselist = B_1 .truelist B.code = B_1 .code
B → true	t_0 = nuevoIndice() B.truelist = crearlista(t_0) B.code = gen('goto' t_0)
B → false	t_0 = nuevoIndice() B.falselist = crearlista(t_0) B.code = gen('goto' t_0)
R → R_1 > R_2	R.lista.true = nuevaLista() R.lista.false = nuevaLista() I= newIndex(), I1 = newIndex() R.lista.true.add(I) R.lista.false.add(I1) R .tipo = max(R_1.tipo, R_2.tipo) t_0= ampliar(R_1.dir, R_1.tipo, R.tipo) t_1= ampliar(R_2.dir, R_2.tipo, R.tipo)

	E.code = gen(E.dir=' t_1' >' t_2') E.code = gen(goto l1)
R → R_1 => R_2	R.lista.true = nuevaLista() R.lista.false = nuevaLista() I= newIndex(), I1 = newIndex() R.lista.true.add(I) R.lista.false.add(I1) R .tipo = max(R_1.tipo, R_2.tipo) t_0= ampliar(R_1.dir, R_1.tipo, R.tipo) t_1= ampliar(R_2.dir, R_2.tipo, R.tipo) E.code = gen(E.dir=' t_1' =>' t_2') E.code = gen(goto l1)
R → R_1 < R_2	R.lista.true = nuevaLista() R.lista.false = nuevaLista() I= newIndex(), I1 = newIndex() R.lista.true.add(I) R.lista.false.add(I1) R .tipo = max(R_1.tipo, R_2.tipo) t_0= ampliar(R_1.dir, R_1.tipo, R.tipo) t_1= ampliar(R_2.dir, R_2.tipo, R.tipo) E.code = gen(E.dir=' t_1' <' t_2') E.code = gen(goto l1)
R → R_1 <= R_2	R.lista.true = nuevaLista() R.lista.false = nuevaLista() I= newIndex(), I1 = newIndex() R.lista.true.add(I) R.lista.false.add(I1) R .tipo = max(R_1.tipo, R_2.tipo) t_0= ampliar(R_1.dir, R_1.tipo, R.tipo) t_1= ampliar(R_2.dir, R_2.tipo, R.tipo) E.code = gen(E.dir=' t_1' <=' t_2') E.code = gen(goto l1)
R → R_1 <> R_2	R.lista.true = nuevaLista() R.lista.false = nuevaLista() I= newIndex(), I1 = newIndex() R.lista.true.add(I) R.lista.false.add(I1) R .tipo = max(R_1.tipo, R_2.tipo) t_0= ampliar(R_1.dir, R_1.tipo, R.tipo) t_1= ampliar(R_2.dir, R_2.tipo, R.tipo) E.code = gen(E.dir=' t_1' <>' t_2') E.code = gen(goto l1)
R → R_1 = R_2	R.lista.true = nuevaLista() R.lista.false = nuevaLista() I= newIndex(), I1 = newIndex() R.lista.true.add(I) R.lista.false.add(I1) R .tipo = max(R_1.tipo, R_2.tipo)

	t_0= ampliar(R_1.dir, R_1.tipo, R.tipo) t_1= ampliar(R_2.dir, R_2.tipo, R.tipo) E.code = gen(E.dir'=' $t_1' =' t_2'$) E.code = gen(goto l1)
R → E	R.tipo = E.tipo R.dir = E.dir
E →E_1 +E_2	E.dir = nuevaTemporal E.tipo = max(E_1.tipo , E_2.tipo) t_1 = ampliar(E_1.dir, E_1.tipo , E.tipo) t_2 = ampliar(E_2.dir, E_2.tipo , E.tipo) E.code = gen(E.dir'=' t_1'+t_2)
E → E_1 * E_2	E.dir = nuevaTemporal E.tipo = max(E_1 .tipo , E_2.tipo) t_1= ampliar(E_1.dir, E_1.tipo , E.tipo) t_2= ampliar(E_2.dir, E_2.tipo , E.tipo) E.code = gen(E.dir'=' t_1'*t_2)
E → E_1 / E_2	E.dir = nuevaTemporal E.tipo = max(E_1 .tipo , E_2.tipo) t_1= ampliar(E_1.dir, E_1.tipo , E.tipo) t_2= ampliar(E_2.dir, E_2.tipo , E.tipo) E.code = gen(E.dir'=' t_1/'t_2')
E →E_1 % E_2	E.dir = nuevaTemporal E.tipo = max(E_1 .tipo , E_2.tipo) t_1= ampliar(E_1.dir, E_1.tipo , E.tipo) t_2= ampliar(E_2.dir, E_2.tipo , E.tipo) E.code = gen(E.dir'=' t_1'%'t_2')
E → (E_1)	E.dir = E_1.dir E.tipo= E_1.tipo
E→ Var	E.dir = nuevaTemporal() E.tipo =Var.tipo E.code = gen()
E → Cad	E.tipo =Cad E.dir=TablaDeCadenas.add(Cad)
E → Char	E.tipo =Char E.dir= TablaDeCadenas.add(Char)
L_a → id [E]	**Si** TS.existe(id) **Entonces** tipo_id = TS.getTipo(id) **Si** TT.getNombre(tipo_id) = array Entonces L_a .dir = nuevaTemporal() L_a.tipo = TT.getTipoBase(tipo_id) L_a .tam = TT.getTam(L_a.tipo) L_a .base = id.dir L_a .code = gen(L_a.dir '=' E.dir '*' L_a .tam) **Sino** error("El id no es un arreglo") **Fin Si** **Sino**

	error(“El id no ha sido declarado”) **Fin Si**
Var_comp → dato_est_sim	**Ya no me dio tiempo de estos perdone :c luego lo completo**
Var_comp → arreglo	
Var_comp → (parámetros)	
dato est sim → dato est sim.id	
dato est sim → ε	
A → (E)	
A → A (E)	
$P_a \rightarrow L_p$	P_a.lista = L_p.lista P_a .num = L_p.num
$P_a \rightarrow$ ε	P_a.lista = nulo P_a.num =0
$L_p \rightarrow L_{p1}$, E	L_p.lista = L_{p1}.lista L_p .lista.append(E.tipo) L_p .num = L_{p1}.num +1