Proyecto Final Compiladores

Definición Dirigida por Sintaxis

|  |  |
| --- | --- |
| Producción | Regla Semántica |
| Programa → declaraciones funciones  P → D F | dir = 0  STS.push(newTS ())  STT.push(newTT ())  P.codigo = F.codigo |
| Declaraciones →  |tipo lista\_var; declaraciones  |tipo\_registro lista\_var; declaraciones  |ϵ  D →T Lv; D  D → Tr Lv; D  D → ϵ | Lv.tipo = T.tipo  Lv.tipo = Tr.tipo |
| Tipo\_registro →  estructura inicio declaraciones fin  Tr → struct {D} | STS.push(newTS ())  STT.push(newTT ())  SDir.push(dir)  dir = 0  TS = STS.pop()  TT = STT.pop()  tam = TS  dir = SDir.pop()  T.tipo=STT.getTop().insert(”struct”,tam,TS ) |
| Tipo → base tipo\_arreglo  T → B Ta | Tipo.type = tipo\_arreglo.type  Base = B.base |
| Base → ent | real | dreal | car | sin  B →ent  B →real  B →dreal  B →car  B →sin | B.base = int  B.base = float  B.base =double  B.base = char  B.base = void |
| Tipo\_arreglo →  |(num) tipo\_arreglo  |ϵ    Ta→ (num) Ta1  Ta→ϵ | Si num.type = ent Entonces  Si num.dir > 0 Entonces  tipo arreglo.type =  STT.getTop().insert(’array’,num,Ta1.tipo)  Sino  error(’...’)  Fin Si  Sino  error(’...’)  Fin Si |
| Lista \_var →  | Lista\_var1, id  | id  Lv→Lv,id  Lv→ id | Si STS.getTop().existe(id) Entonces  STS.getTop().insert(id, typeGBL, dir, ’var’, null, null)  dir dir + STT.getTop().getTam(typeGBL)  Sino  error(’el id no esta definido’)  Fin Si |
| Funciones →  | def tipo id(argumentos) inicio declaraciones sentencias fin funciones  | ϵ  F→ define T id (A) {D Ss}  F→ ϵ | Si no STS.getTop().existe(id) Entonces  STS.push(newTS())  STT.push(newTT())  SDir.push(dir)  dir = 0  listaRET = newListRet()  Si cmpRet(lista retorno, tipo.type) Entonces  L = newLabel()  backpatch(sentencias.nextlist, L)  genCode(label L)  STS.pop()  STT.pop()  Sino  error(’Los tipos no coinciden’)  Fin Si  Sino  error(’El id ya existe’)  Fin Si |
| Argumentos →  | |  |
| Lista\_arg → |  |
| Arg → |  |
| Tipo\_arg → |  |
| Param\_arr |  |
| Sentencias → |  |
| Sentencia → |  |
| Casos → |  |
| Predeterminado → |  |
| E\_bool → |  |
| Relacional → |  |
| Expresion → |  |
| Variable → |  |
| Variable\_comp → |  |
| Dato\_est\_sim → |  |
| Arreglo → |  |
| Parametros → |  |
| Lista\_param → |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Símbolo | Nombre | Símbolo |
| Programa | P | Sentencias | Ss |
| Declaraciones | D | Sentencia | S |
| Tipo registro | Tr | Casos | C |
| Tipo | T | Predeterminado | Pre |
| Base | B | E\_bool | Eb |
| Tipo\_arreglo | Ta | Relacional | R |
| Lista \_var | Lv | Expresion | E |
| Funciones | F | Variable | V |
| Argumentos | A | Variable\_comp | Vc |
| Lista\_arg | La | Dato\_est\_sim | Des |
| Arg | Arg | Arreglo | Ar |
| Tipo\_arg | Targ | Parametros | Par |
| Param\_arr | Pa | Lista\_param | Lp |