

Gramática de Atributos acondicionada para la construcción de la Tabla de Símbolos (2º Cuatrimestre)

Prog ::= Cabecera Decs Bloque

Decs.tsph = Cabecera.tsph

Decs.dirsh = Cabecera.dirs

Decs.nivelh = Cabecera.nivel

Bloque.tsph = Decs.tsph

Bloque.nivelh = Cabecera.nivel

Cabecera ::= PROGRAM id PYCOMA

Cabecera.dirs = 0

Cabecera.nivel = 0

Cabecera.tsph = añadeTablaSimbolos(nuevaPilaTS(), Cabecera.nivel, nuevaTS())

Cabecera.tsph = añadeID(Cabecera.tsph, id.lex, Cabecera.nivel, < tipo: tipoError, clase: cabecera >)

Cabecera.dirs = añadePilaDireccion(nuevaPilaDir(), Cabecera.nivel, 0)

Sección de declaraciones

Decs ::= DTipos RDecs

DTipos.tsph = Decs.tsph

RDecs.tsph = DTipos.tsph

Decs.tsph = RDecs.tsph

DTipos.dirsh = Decs.dirsh

RDecs.dirsh = Decs.dirsh

Decs.dirs = RDecs.dirs

DTipos.nivelh = Decs.nivelh

RDecs.nivelh = Decs.nivelh

RDecs ::= λ

RDecs.tsph = RDecs.tsph

RDecs.dirs = RDecs.dirsh

RDecs ::= Procs

Procs.tsph = RDecs.tsph

Procs.dirsh = RDecs.dirsh

RDecs.dirs = Procs.dirs

Procs.nivelh = RDecs.nivelh

RDecs ::= Vars RDecs2

Vars.tsph = RDecs.tsph

RDecs2.tsph = Vars.tsph

RDecs.tsph = RDecs2.tsph

Vars.dirsh = RDecs.dirsh

RDecs2.dirsh = Vars.dirs
RDecs.dirs = RDecs2.dirs
Vars.nivelh = RDecs.nivelh
RDecs2.nivelh = RDecs.nivelh
RDecs2 ::= Procs
Procs.tsph = RDecs2.tsph
Procs.dirsh = RDecs2.dirsh
RDecs2.dirs = Procs.dirs
Procs.nivelh = RDecs2.nivelh

RDecs2 ::= λ
RDecs2.tsph = RDecs2.tsph
RDecs2.dirs = RDecs2.dirsh

Decs ::= Vars RDecs3
Vars.tsph = Decs.tsph
RDecs3.tsph = Vars.tsph
Decs.tsph = RDecs3.tsph
Vars.dirsh = RDecs.dirsh
RDecs3.dirsh = Vars.dirs
Decs.dirs = RDecs3.dirs
Vars.nivelh = RDecs.nivelh
RDecs2.nivelh = RDecs.nivelh

RDecs3 ::= Procs
Procs.tsph = RDecs3.tsph
Procs.dirsh = RDecs3.dirsh
RDecs3.dirs = Procs.dirs
Procs.nivelh = RDecs3.nivelh

RDecs3 ::= λ
RDecs2.tsph = RDecs2.tsph
RDecs2.dirs = RDecs2.dirsh

Decs ::= λ
Decs.tsph = Decs.tsph
Decs.dirs = Decs.dirsh

Declaración de tipos

DTipos ::= SECTIPOS RTipos
RTipos.tsph = DTipos.tsph
DTipos.tsph = RTipos.tsph
Rtipos.nivelh = DTipos.nivelh

RTipos ::= RTipos2 RTipos
RTipos2.tsph = RTipos0.tsph
RTipos1.tsph = RTipos.tsph
RTipos0.tsph = RTipos1.tsph
RTipos2.nivelh = RTipos0.nivelh

RTipos1.nivelh = RTipos0.nivelh

RTipos ::= λ
RTipos.tsph = RTipos.tsph
RTipos.nivel = RTipos.nivelh

RTipos2 ::= id IGUAL Tipo PYCOMA
RTipos.tsph = RTipos.tsph
RTipos.nivel = RTipos.nivelh

Tipo ::= TIPENT
Tipo.tipo = <t: Entero, tam: 1>

Tipo ::= TIPBOOL
Tipo.tipo = <t: Boolean, tam: 1>

Tipo ::= id
Tipo.tipo = <t: Tipo.tsph[id.lex].tipo.t, id: id.lex, tam: Tipo.tsph[id.lex].tipo.tam>

Tipo ::= TPUNTERO Tipo
Tipo1.tsph = Tipo0.tsph
Tipo0.tipo = <t: Puntero, tBase: Tipo1.tipo, tam: 1>

Tipo ::= TARRAY [0..numero] of Tipo
Tipo1.tsph = Tipo0.tsph
Tipo0.tipo = <t: Array, numElems: valorDe(numero.lex), tBase: Tipo1.tipo, tam: (valorDe(numero.lex)+1)* Tipo1.tipo.tam>

Declaración de variables

***Vars ::= VAR Tvar2
Tvar2.tsph = Vars.tsph
Vars.tsp = Tvar2.tsp
Tvar2.nivelh = Vars.nivelh
Tvar2.dirsh = Vars.dirsh
Vars.dirs = Tvar2.dirs***

***Tvar2 ::= λ
Tvar2.tsp = RTvar2.tsph
Tvar2.dirs = Tvar2.dirsh
Tvar2.nivel = Tvar2.nivelh***

***Tvar2 ::= id RTvar2
Tvar2.tsp = añadeVariable(RTvar2.tsp, id.lex, <direccion:RTvar2.dirs, tipo:
RTvar2.tipo, clase: variable, tam: RTvar2.tipo.tam, nivel: Tvar2.nivelh >)
RTvar2.dirsh=Tvar2.dirsh
RTvar2.tsph = Tvar2.tsph
Tvar2.tipo = RTvar2.tipo
RTvar2.nivelh = Tvar2.nivelh
Tvar2.dirs = incrementaDireccion(RTvar2.dirs, RTvar2.nivelh, RTvar2.tipo.tam)***

***RTvar2 ::= COMA RTvar2
RTvar2₁.dirsh=RTvar2₀.dirsh
RTvar2₁.tsph = RTvar2₀.tsph
RTvar2₀.tipo = RTvar2₁.tipo
RTvar2₁.nivelh = RTvar2₀.nivelh
RTvar2₀.tsp = RTvar2₁.tsp
RTvar2₀.dirs = RTvar2₁.dirs***

***RTvar2 ::= 2PUNTOS Tipos PYCOMA
Tipos.tsph = RTvar2.tsph
RTvar2.tipo = Tipos.tipo***

***Tipos ::= TIPENT
Tipos.tipo = <t: Entero, tam: 1>***

***Tipos ::= TIPBOOL
Tipos.tipo = <t: Boolean, tam: 1>***

***Tipos ::= id
Tipos.tipo = <t: Tipos.tsph[id.lex].tipo.t, id: id.lex, tam: Tipos.tsph[id.lex].tipo.tam>***

Declaración de procedimientos

Procs ::= TProc Procs
TProc.tsph = Procs0.tsph
Procs1.tsph = TProc.tsph
TProc.dirsh = Procs0.dirsh
Procs1.dirsh = incrementaDireccion(Procs0.dirsh, Procs1.nivelh, TProc.dirs)
Procs0.dirs = Procs1.dirs
TProc.nivelh = Procs0.nivelh
Procs1.nivelh = Procs0.nivelh
Procs0.nivel = maximo(Procs1.nivel, TProc.nivel)

Procs ::= λ
Procs.tsp = Procs.tsph
Procs.dirs = Procs.dirs

TProc ::= PROC id Params PYCOMA BloqProc
Params.tsph = TProc.tsph
Params.dirsh = TProc.dirsh
Params.nivel = TProc.nivelh + 1
BloqProc.nivel = TProc.nivelh + 1
BloqProc.tsph = añadeID(Params.tsp, id.lex, <clase: proc, tipo: <t: proc, params: Params.params>, nivel: BloqProc.nivel>)
BloqProc.dirs = Params.dirs
TProc.tsp = añadeID(TProc.tsph, id.lex, <clase: proc, tipo: <t: proc, params: Params.params>, nivel: TProc.nivelh>)
TProc.dirs = BloqProc.dirs
TProc.nivel = BloqProc.nivel

Params ::= PA ListaParams PC
ListaParams.tsph = Params.tsph
ListaParams.nivelh = Params.nivelh
Params.tsp = ListaParams.tsp
ListaParams.dirsh = Params.dirs
Params.dirs = ListaParams.dirs
Params.params = ListaParams.params

Params ::= λ
Params.tsp = Params.tsph
Params.dirs = Params.dirsh
Params.params = []

ListaParams ::= Params2 RListaParams
Params2.tsph = ListaParams.tsph
RListaParams.nivelh = ListaParams.nivelh
Params2.nivelh = ListaParams.nivelh
Params2.dirsh = ListaParams.dirsh
RListaParams.dirsh = Params2.dirs
RListaParams.tsph = Params2.tsp
ListaParams.tsp = RListaParams.tsp

RListaParams.dirs = *Params2.dirs*
ListaParams.dirs = *RListaParams.dirs*
ListaParams.nparams = *Params2.nparams* + *RListaParams.nparams*
ListaParams.params = *RListaParams.params* ++ [*Params2.params*]
Params2.mod0 = valor

RListaParams ::= PYCOMA *ListaParams*
ListaParams.tsph = *RListaParams.tsph*
ListaParams.dirsh = *RListaParams.dirsh*
ListaParams.nivelh = *RListaParams.nivelh*
RListaParams.tsp = *ListaParams.tsp*
RListaParams.dirs = *ListaParams.dirs*
RListaParams.nparams = *ListaParams.nparams*
RListaParams.params = *ListaParams.params*

RListaParams ::= λ
RListaParams.tsph = *RListaParams.tsph*
RListaParams.dirs = *RListaParams.dirsh*
RListaParams.nparams = 0
RListaParams.params = []

ListaParams ::= TVAR *RListaParams2*
RListaParams2.dirsh = *ListaParams.dirsh*
RListaParams2.tsph = *ListaParams2.tsph*
RListaParams2.nivelh = *ListaParams2.nivelh*
ListaParams.tsp = *RListaParams2.tsp*
ListaParams.dirs = *RListaParams2.dirs*
ListaParams.nparams = *RListaParams2.nparams*
ListaParams.params = *RListaParams2.params*

RListaParams2 ::= *Params2*
Params2.tsph = *RListaParams2.tsph*
Params2.nivelh = *RListaParams2.nivelh*
Params2.dirsh = 0
RListaParams2.tsp = *Params2.tsp*
RListaParams2.dirs = *Params2.dirs*
RListaParams2.nparams = *Params2.params*
RListaParams2.params = [*Params2.params*]
Params2.mod0 = variable

RListaParams2 ::= *Params2* PYCOMA *ListaParams*
Params2.tsph = *ListaParams0.tsph*
ListaParams1.nivelh = *ListaParams0.nivelh*
Params2.nivelh = *ListaParams0.nivelh*
Params2.dirsh = *ListaParams0.dirsh*
ListaParams1.dirsh = *Params2.dirs*
ListaParams1.tsph = *Params2.tsp*
ListaParams0.tsp = *ListaParams1.tsp*
ListaParams1.dirs = *Params2.dirs*
ListaParams0.dirs = *ListaParams1.dirs*

ListaParams0.nparams = *Params2.nparams* + *ListaParams1.params*
ListaParams0.params = *ListaParams1.params* ++ [*Params2.params*]
Params2.modos = *valor*

Params2 ::= id *RParams2*
RParams2.tsph = *Params2.tsph*
RParams2.dirsh = *Params2.dirsh*
RParams2.nivelh = *Params2.nivelh*
Params2.tsp = *añadeID(RParams2.tsp, id.lex, <direccion: RParams2.dirs, tipo:*
RParams2.tipo, clase: if Params2.modos = variable then pVariable else valor,
tam: if Params2.modos = variable then 1 else RParams2.tipo.tam, nivel:
Params2.nivelh, modos: Params2.modos >)
Params2.nparams = 1 + *RParams2.nparams*
Params2.params = *añadeParametro(RParams2.tsp[id.lex], Params2.nparams)*
Params2.dirs = *incrementaDireccion(RParams2.dirs, Params2.nivelh,*
if Params2.modos = variable then 1 else RParams2.tipo.tam)

RParams2 ::= COMA *Params2*
Params2.tsph = *RParams2.tsph*
RParams2.tsp = *Params2.tsp*
Params2.dirsh = *RParams2.dirsh*
Rparams2.dirs = *Params2.dirs*
Params2.nivelh = *RParams2.nivelh*
RParams2.nparams = *Params2.nparams*

RParams2 ::= 2PUNTOS *Tipos*
Tipos2.tsph = *RParams2.tsph*
RParams2.nparams = 0

BloqProc ::= Decs2 *Bloque*
Decs2.tsph = *BloqProc.tsph*
Bloque.tsph = *BloqProc.tsph*
Decs2.nivelh = *BloqProc.nivel*
Bloque.nivelh = *BloqProc.nivelh*
Decs2.dirsh = *BloqProc.dirsh*
BloqProc.tsp = *Decs2.tsp*
BloqProc.dirs = *Decs2.dirs*
Bloque.dirsh = *Decs2.dirs*

Decs2 ::= Vars
Decs2.tsp = *Decs2.tsph*
Decs2.dirs = *Decs2.dirsh*

Decs2 ::= λ
Vars.tsph = *Decs2.tsph*
Vars.dirsh = *Decs2.dirsh*
Vars.nivelh = *Decs2.nivelh*
Decs2.dirs = *Vars.dirs*
Decs2.tsp = *Vars.tsp*

Cuerpo del programa

Bloque ::= INICIO TBloque2 FIN
TBloque2.tsph = Bloque.tsph
TBloque2.nivelh = Bloque.nivelh

TBloque2 ::= TSentencia TBloque2
TSentencia.tsph = TBloque2₀.tsph
TBloque2₁.tsph = TBloque2₀.tsph
TSentencia.nivelh = TBloque2₀.nivelh
TBloque2₁.nivelh = TBloque2₀.nivelh

TBloque2 ::= λ

TSentencia ::= TAsig
TAsig.tsph = TSentencia.tsph
TAsig.nivelh = TSentencia.nivelh

TSentencia ::= TRead
TRead.tsph = TSentencia.tsph
TRead.nivelh = TSentencia.nivelh

TSentencia ::= TWrite
TWrite.tsph = TSentencia.tsph
TWrite.nivelh = TSentencia.nivelh

TSentencia ::= TNPunt
TNPunt.tsph = TSentencia.tsph
TNPunt.nivelh = TSentencia.nivelh

TSentencia ::= TLiberar
TLiberar.tsph = TSentencia.tsph
TLiberar.nivelh = TSentencia.nivelh

TSentencia ::= TLLamadaProc
TLLamadaProc.tsph = TSentencia.tsph
TLLamadaProc.nivelh = TSentencia.nivelh

TSentencia ::= TIf
TIf.tsph = TSentencia.tsph
TIf.nivelh = TSentencia.nivelh

TSentencia ::= TWhile
Exp.tsph = TWhile.tsph
TBloque2.tsph = TWhile.tsph
Exp.nivelh = TWhile.nivelh
TBloque2.nivelh = TWhile.nivelh

Tif ::= SI PA Exp PC ENTONCES INICIO TBloque2 FIN RTif

Exp.tsph = TIf.tsph

TBloque2.tsph = TIf.tsph

Exp.nivelh = TIf.nivelh

TBloque2.nivelh = TIf.nivelh

RTif.tsph = TIf.tsph

RTif.nivelh = TIf.nivelh

RTif ::= λ

RTif ::= SINO INICIO TBloque2 FIN

TBloque2.tsph = TIf.tsph

TBloque2.nivelh = TIf.nivelh

TWhile ::= MIENTRAS PA Exp PC HACER INICIO TBloque2 FIN

Exp.tsph = TWhile.tsph

TBloque2.tsph = TWhile.tsph

Exp.nivelh = TWhile.nivelh

TBloque2.nivelh = TWhile.nivelh

TLLamadaProc ::= id PA Params3 PC PYCOMA

Params3.tsph = TLLamadaProc.tsph

Params3.nivelh = TLLamadaProc.nivelh

Params3 ::= ListaParams3

ListaParams3.tsph = Params3.tsph

ListaParams3.nivelh = Params3.nivelh

Params3.nparams = ListaParams3.nparams

Params3 ::= λ

ListaParams3.nparams = 0

ListaParams3 ::= Exp RListaParams3

ListaParams3.nparams = 1 + RListaParams3.nparams

Exp.tsph = ListaParams3.tsph

Exp.nivelh = ListaParams3.nivelh

RListaParams3.tsph = ListaParams3.tsph

RListaParams3.nivelh = ListaParams3.nivelh

RListaParams3 ::= COMA ListaParams3

RListaParams3.nparams = ListaParams3.nparams

ListaParams3.tsph = RListaParams3.tsph

ListaParams3.nivelh = RListaParams3.nivelh

RListaParams3 ::= λ

RListaParams3.nparams = 0

TRead ::= LEER PA id PC PYCOMA

TWrite ::= ESCRIBIR PA id PC PYCOMA

TNPunt ::= NUEVO PA id PC PYCOMA

TLiberar ::= LIBERAR PA id PC PYCOMA

TAsig ::= Descriptor ASIG Exp

Descriptor.tsph = TAsig.tsph

Descriptor.nivelh = TAsig.nivelh

Exp.tsph = TAsig.tsph

Exp.nivelh = TAsig.nivelh

Descriptor ::= Descriptor2

Descriptor2.tsph = Descriptor2.tsph

Descriptor2.nivelh = Descriptor2.nivelh

Descriptor2 ::= id

Descriptor2 ::= Descriptor2[Exp]

Descriptor2₁.tsph = Descriptor2₀.tsph

Descriptor2₁.nivelh = Descriptor2₀.nivelh

Exp.tsph = Descriptor2₀.tsph

Exp.nivelh = Descriptor2₀.nivelh

Descriptor2 ::= ^Descriptor2

Descriptor2₁.tsph = Descriptor2₀.tsph

Descriptor2₁.nivelh = Descriptor2₀.nivelh

Exp ::= ExpSum RExp

RExp.tsph = Exp.tsph

ExpSum.tsph = Exp.tsph

ExpSum.nivelh = Exp.nivelh

RExp.nivelh = Exp.nivelh

Exp.moda = asigna_modo(ExpSum.moda, RExp.moda)

RExp ::= OpRel Exp

Exp₁.tsph = Exp₀.tsph

ExpSum.tsph = Exp₀.Tsph

Exp.moda = valor

RExp ::= λ

RExp.moda = variable

ExpSum ::= ExpProd RExpSum

RExpSum.tsph = ExpSum.tsph

ExpProd.tsph = ExpSum.tsph

ExpProd.nivelh = ExpSum.nivelh

RExpSum.nivelh = ExpSum.nivelh

ExpSum.moda = asigna_modo(RExpSum.moda, ExpProd.moda)

RExpSum ::= OpAd ExpSum
RExpSum.tsph = ExpSum.tsph
RExpSum.nivelh = ExpSum.nivelh
RExpSum.mod0 = valor

RExpSum ::= OR ExpSum
ExpSum.tsph = RExpSum.tsph
ExpSum.nivelh = RExpSum.nivelh
RExpSum.mod0 = valor

RExpSum ::= λ
RExpSum.mod0 = variable

ExpProd ::= ExpFact RExpProd
ExpFact.tsph = ExpProd.tsph
RExpProd.tsph = ExpProd.tsph
ExpFact.nivelh = ExpProd.nivelh
RExpProd.nivelh = ExpProd.nivelh
ExpProd.mod0 = asigna_mod0(ExpFact.mod0, RExpProd.mod0)

RExpProd ::= OpProd ExpProd
ExpProd.tsph = RExpProd.tsph
ExpProd.nivelh = RExpProd.nivelh
RExpProd.mod0 = valor

RExpProd ::= AND ExpProd
ExpProd.tsph = RExpProd.tsph
ExpProd.nivelh = RExpProd.nivelh
RExpProd.mod0 = valor

RExpProd ::= λ
RExpProd.mod0 = variable

ExpFact ::= (Exp)
Exp.tsph = ExpFact.tsph
Exp.nivelh = ExpFact.nivelh
ExpFact.mod0 = Exp.mod0

ExpFact ::= OpAd ExpFact
ExpFact1.tsph = ExpFact0.tsph
ExpFact.mod0 = valor

ExpFact ::= numero
ExpFact.mod0 = valor

ExpFact ::= True
ExpFact.mod0 = valor

ExpFact ::= False
ExpFact.mod0 = valor

ExpFact ::= Not ExpFact
ExpFact₁.Tsph = ExpFact₀.Tsph
ExpFact₀.modo = valor

ExpFact ::= Descriptor
Descriptor.tsph = ExpFact.tsph
ExpFact.modo = variable