Gramática de Atributos para la construcción de la Tabla de Símbolos 2º Cuatrimestre

En este documento se presenta la gramática de atributos empleada para formalizar la construcción de la tabla de símbolos. Primeramente hablaremos de los atributos semánticos y operaciones posibles en la tabla de símbolos, para posteriormente dar paso a la gramática propiamente dicha.

Ante la inclusión de ampliaciones al lenguaje del primer cuatrimestre, y debido a la mala previsión a la hora de diseñar la sintaxis, la gramática obtenida en aquel momento nos presentaba serios problemas. Como consecuencia de ello, hemos tenido que rediseñar en gran medida las categorías sintácticas, resultando una gramática bastante diferente.

En cuanto a atributos, debido a la inclusión de procedimientos, la primera modificación es un "contenedor de tablas de símbolos" (atributo ***tsp***), (que instintivamente se puede interpretar a modo de estructura pila accediendo a la tabla que nos indique el nivel en cada caso), en el que se irán almacenando las tablas de símbolos asociadas a su nivel de declaración. Así pues tenemos un nuevo atributo ***nivel***, heredado, para indicar el nivel de declaración de los componentes, así como permitir el acceso a la tabla de símbolos apropiada del contenedor. También disponemos de un atributo ***dirs,*** que será un contenedor de direcciones, en el que aparecerán las direcciones de declaración asociadas cada tabla de símbolos existente. Aparecen además atributos sintetizados completamente nuevos como ***clase***, ***tipo***, ***tam***, ***modo*** y ***params.*** Estos nuevos atributos representan los punteros a la información de la clase de identificador, el tipo, el tamaño, y por último modo y params usados para almacenar y diferenciar el paso por parámetros por valor o variable respectivamente, información necesaria a la hora de manejar procedimientos.

Continuamos con una descripción de las operaciones de la tabla de símbolos, definiendo la cabecera de dichas operaciones, así como describiendo informalmente su cometido, incluyendo el propósito de cada uno de sus parámetros:

añadeTablaSimbolos(pila_de_tablas,nivel,ts): Añade la tabla de símbolos *ts* al contenedor de tablas de símbolos, asociada al nivel indicado.
añadeDireccion(contenedor, nivel, dir): Añade la dirección *dir* al contenedor de direcciones, asociada al nivel indicado.
incrementaDireccion(contenedor, nivel, cantidad): Incrementa la dirección asociada al nivel indicado, un valor *cantidad*.
añadeID(t: TS, id: Identificador, ps.Propiedades): TS: El resultado es la tabla de símbolos que resulta de añadir a la tabla *t* el identificador *id* con la direccion *dir*.
existeID(t: TS, id: Identificador): Boolean: El resultado es true si *id* aparece en *t*, y falso en otro caso.
valorDe(v: String): {Boolean, Integer} Nos devuelve el valor del v.

Estructura general

Prog ::= Cabecera Decs Bloque
Decs.tsph = Cabecera.tsp
Decs.dirsh=Cabecera.dirs
Decs.nivelh = Cabecera.nivel
Bloque.tsph = Decs.tsp
Bloque.nivelh = Cabecera.nivel

Cabecera ::= PROGRAM id PYCOMA
Cabecera.dirs = 0
Cabecera.nivel = 0
Cabecera.tsp = añadeTablaSimbolos(nuevaPilaTS(), Cabecera.nivel, nuevaTS())
Cabecera.tsp = añadeID(Cabecera.tsp, id.lex, Cabecera.nivel,< tipo: tipoError, clase: cabecera >)
Cabecera.dirs = añadePilaDireccion(nuevaPilaDir(), Cabecera.nivel, 0)

Sección de declaraciones

Decs ::= DTipos Vars Procs
DTipos.tsph = Decs.tsph
Vars.tsph = DTipos.tsp
Procs.tsph = Vars.tsp
Decs.tsp =Procs.tsp
DTipos.dirsh = Decs.dirsh
Vars.dirsh = DTipos.dirs
Procs.dirsh = Vars.dirs
Decs.dirs =Procs.dirs
DTipos.nivelh = Decs.nivelh
Vars.nivelh = Decs.nivelh
Procs.nivelh = Decs.nivelh

Decs ::= DTipos Vars
DTipos.tsph = Decs.tsph
Vars.tsph = DTipos.tsp
Decs.tsp = Vars.tsp
DTipos.dirsh = Decs.dirsh
Vars.dirsh = DTipos.dirs
Decs.dirs = Vars.dirs
DTipos.nivelh = Decs.nivelh
Vars.nivelh = Decs.nivelh

Decs ::= DTipos Procs
DTipos.tsph = Decs.tsph
Procs.tsph = DTipos.tsp
Decs.tsp = Procs.tsp
DTipos.dirsh = Decs.dirsh
Procs.dirsh = DTipos.dirs
Decs.dirs = Procs.dirs
DTipos.nivelh = Decs.nivelh
Procs.nivelh = Decs.nivelh

Decs ::= Vars Procs
Vars.tsph = Decs.tsph
Procs.tsph = Vars.tsp
Decs.tsp = Procs.tsp
Vars.dirsh = Decs.dirsh
Procs.dirsh = Vars.dirs
Decs.dirs = Procs.dirs
Vars.nivelh = Decs.nivelh
Procs.nivelh = Decs.nivelh

Decs ::= Vars
Vars.tsph = Decs.tsph
Decs.tsp = Vars.tsp
Vars.dirsh = Decs.dirsh
Decs.dirs = Vars.dirs
Vars.nivelh = Decs.nivelh

Decs ::= Procs
Procs.tsph = Decs.tsph
Decs.tsp = Procs.tsp
Procs.dirsh = Decs.dirsh
Decs.dirs = Procs.dirs
Procs.nivelh = Decs.nivelh

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

Declaración de tipos

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

RTipos ::= RTipos2 RTipos
RTipos2.tsph = RTipos$_0$.tsph
RTipos$_1$.tsph = RTipos.tsp
RTipos$_0$.tsp = RTipos$_1$.tsp
RTipos2.nivelh = RTipos$_0$.nivelh
RTipos$_1$.nivelh = RTipos$_0$.nivelh
RTipos ::= λ
RTipos.tsp = RTipos.tsph
RTipos.nivel = RTipos.nivelh

RTipos2 ::= id IGUAL Tipo PYCOMA
Tipo.tsph = RTipos2.tsph
RTipos2.tsp = añadeTipoConstruido(RTipos2.tsph[RTipos2.nivelh], id.lex, < tipo: Tipo.tipo.t, clase: tipo, tam: Tipo.tipo.tam, nivel:RTipos2.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
Tipo$_1$.tsph = Tipo$_0$.tsph
Tipo$_0$.tipo = <t: Puntero, tBase: Tipo$_1$.tipo, tam: 1>

Tipo ::= TARRAY [0..numero] of Tipo
Tipo$_1$.tsph = Tipo$_0$.tsph
Tipo$_0$.tipo = <t: Array, numElems: valorDe(numero.lex), tBase: Tipo$_1$.tipo, tam: (valorDe(numero.lex)+1)* Tipo$_1$.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 ::= RTvar2 Tvar2
RTvar2.tsph = Tvar2$_0$.tsph
Tvar2$_1$.tsph = RTvar2.tsp
Tvar2$_0$.tsp = Tvar2$_1$.tsp
RTvar2.dirsh = Tvar2$_0$.dirsh
Tvar2$_1$.dirsh = RTvar2.dirs
Tvar2$_0$.dirs = Tvar2$_1$.dirs
RTvar2.nivelh = Tvar2$_0$.nivelh
Tvar2$_1$.nivelh = Tvar2$_0$.nivelh

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

RTvar2 ::= id COMA RTvar2
RTvar2$_0$.tsp = añadeVariable(RTvar2$_1$.tsp, id.lex, <direccion:RTvar2$_1$.dirs, tipo: RTvar2$_1$.tipo, clase: variable, tam: RTvar2$_1$.tipo.tam, nivel: RTvar2$_0$.nivelh >)
RTvar2$_1$.dirsh=RTvar2$_0$.dirsh
RTvar2$_1$.tsph = RTvar2$_0$.tsph
RTvar2$_0$.tipo = RTvar2$_1$.tipo
RTvar2$_1$.nivelh = RTvar2$_0$.nivelh
RTvar2$_0$.dirs = incrementaDireccion(RTvar2$_1$.dirs, RTvar2$_1$.nivelh, RTvar2$_1$.tipo.tam)

RTvar2 ::= id 2PUNTOS Tipos PYCOMA
Tipos.tsph = RTvar2.tsph
RTvar2.tsp = añadeVariable(RTvar2.tsph, id.lex, <direccion:RTvar2.dirsh, tipo: Tipos.tipo, clase: variable, tam: Tipo.tipo.tam, nivel: RTvar2.nivelh >)
RTvar2.tipo = Tipos.tipo
RTvar2.dirs = incrementaDireccion(RTvar2.dirsh, RTvar2.nivelh, Tipos.tipo.tam)

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 ::= TDProc Procs
TDProc.tsph = Procs$_0$.tsph
Procs$_1$.tsph =TDProc.tsph
TDProc.dirsh = Procs$_0$.dirsh
Procs$_1$.dirsh = incrementaDireccion(Procs$_0$.dirsh, Procs$_1$.nivelh, TDProc.dirs)
Procs$_0$.dirs = Procs$_1$.dirs
TDProc.nivelh = Procs$_0$.nivelh
Procs$_1$.nivelh = Procs$_0$.nivelh
Procs$_0$.nivel = maximo(Procs$_1$.nivel, TDProc.nivel)

Procs ::= TDProc
TDProc.tsph = Procs.tsph
Procs.tsp = TDProc.tsp
TDProc.dirsh = Procs.dirsh
Procs.dirs = TDProc.dirs
TDProc.nivelh = Procs.nivelh

TDProc ::= PROC id Params PYCOMA BloqProc
Params.tsph= TDProc.tsph
Params.dirsh = TDProc.dirsh
Params.nivel= TDProc.nivelh+1
BloqProc.nivel= TDProc.nivelh+1
BloqProc.tsph= añadeID(Params.tsp, id.lex, <clase: proc, tipo: <t: proc, params: Params.params>, nivel: BloqProc.nivel>)
BloqProc.dirs= Params.dirs
TDProc.tsp= añadeID(TDProc.tsph, id.lex, <clase: proc, tipo: <t: proc, params: Params.params>, nivel: TDProc.nivelh>)
TDProc.dirs= BloqProc.dirs
TDProc.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
Params2.tsph= ListaParams.tsph
Params2.nivelh= ListaParams.nivelh
Params2.dirsh= 0
ListaParams.tsp= Params2.tsp
ListaParams.dirs= Params2.dirs
ListaParams.nparams= Params2.nparams
ListaParams.params=[Params2.params]
Params2.modo = valor

ListaParams ::= TVAR Params2
Params2.tsph= ListaParams.tsph
Params2.nivelh= ListaParams.nivelh
Params2.dirsh= 0
ListaParams.tsp= Params2.tsp
ListaParams.dirs= Params2.dirs
ListaParams.nparams= Params2.nparams
ListaParams.params=[Params2.params]
Params2.modo = variable

ListaParams ::= Params2 PYCOMA ListaParams
Params2.tsph = ListaParams$_0$.tsph
ListaParams$_1$.nivelh= ListaParams$_0$.nivelh
Params2.nivelh= ListaParams$_0$.nivelh
Params2.dirsh= ListaParams$_0$.dirsh
ListaParams$_1$.dirsh= Params2.dirs
ListaParams$_1$.tsph = Params2.tsp
ListaParams$_0$.tsp = ListaParams$_1$.tsp
ListaParams$_1$.dirs= Params2.dirs
ListaParams$_0$.dirs= ListaParams$_1$.dirs
ListaParams$_0$.nparams= Params2.nparams + ListaParams$_1$.params
ListaParams$_0$.params= ListaParams$_1$.params ++ [Params2.params]
Params2.modo = valor

ListaParams ::= TVAR Params2 PYCOMA ListaParams
Params2.tsph = ListaParams$_0$.tsph
ListaParams$_1$.nivelh= ListaParams$_0$.nivelh
Params2.nivelh= ListaParams$_0$.nivelh
Params2.dirsh= ListaParams$_0$.dirsh
ListaParams$_1$.dirsh= Params2.dirs
ListaParams$_1$.tsph = Params2.tsp
ListaParams$_0$.tsp = ListaParams$_1$.tsp
ListaParams$_1$.dirs= Params2.dirs
ListaParams$_0$.dirs= ListaParams$_1$.dirs
ListaParams$_0$.nparams= Params2.nparams + ListaParams$_1$.params
ListaParams$_0$.params= ListaParams$_1$.params ++ [Params2.params]
Params2.modo = variable

Params2 ::= id COMA Params2
Params2$_0$.tsp=añadeID(Params2$_1$.tsp, id.lex, <direccion: Params2$_1$.dirs, tipo: Params2$_1$.tipo, clase: if Params2$_0$.modo = variable then pVariable else valor, tam: if Params2$_0$.modo = variable then 1 else Params2$_1$.tipo.tam, nivel: Params2$_0$.nivelh, modo: Params2$_0$.modo >)
Params2$_0$.nparams = 1 + Params2$_1$.nparams
Params2$_0$.params = añadeParametro(Params2$_1$.tsp[id.lex], Params2$_0$.nparams)
Params2$_0$.dirs=incrementaDireccion(Params2$_1$.dirs, Params2$_0$.nivelh, if Params2$_0$.modo = variable then 1 else Params2.tipo.tam)

Params2 ::= id 2PUNTOS Tipos
Params2.tsp=añadeID(creaTS(Params2.tsph), id.lex, <direccion: Params2.dirsh, tipo: Tipos.tipo, clase: if Params2.modo = variable then pVariable else valor, tam: if Params2.modo = variable then 1 else Tipos.tipo.tam, nivel: Params2.nivelh, modo: Params2.modo>)
Params2.nparams = 1
Params2.dirs=incrementaDireccion(Params2.dirsh, Params2.nivelh, if Params2.modo = variable then 1 else Tipos.tipo.tam)
Params2.tipo =Tipos.tipo

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 ::= λ
Decs2.tsp = Decs2.tsph
Decs2.dirs = Decs2.dirsh

Decs2 ::= Vars
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 ::= λ

TBloque2 ::= TSentencia TBloque2
TSentencia.tsph = TBloque2$_0$.tsph
TBloque2$_1$.tsph = TBloque2$_0$.tsph
TSentencia.nivelh = TBloque2$_0$.nivelh
TBloque2$_1$.nivelh = TBloque2$_0$.nivelh

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

TIf ::= SI PA Exp PC ENTONCES INICIO TBloque2 FIN SINO INICIO TBloque2 FIN
Exp.tsph = TIf.tsph
TBloque2$_0$.tsph = TIf.tsph
TBloque2$_1$.tsph = TIf.tsph
Exp.nivelh = TIf.nivelh
TBloque2$_0$.nivelh = TIf.nivelh
TBloque2$_1$.nivelh = TIf.nivelh

TIf ::= SI PA Exp PC ENTONCES INICIO TBloque2 FIN
Exp.tsph = TIf.tsph
TBloque2.tsph = TIf.tsph
Exp.nivelh = TIf.nivelh
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

TSentencia ::= TLlamadaProc
TLlamadaProc.tsph = TSentencia.tsph
TLlamadaProc.nivelh = TSentencia.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
Exp.tsph = ListaParams3.tsph
Exp.nivelh = ListaParams3.nivelh
ListaParams3.nparams = 1

ListaParams3 ::= Exp COMA ListaParams3
$ListaParams3_0.nparams = 1 + ListaParams3_1.nparams$
$ListaParams3_1.tsph = ListaParams3_0.tsph$
$ListaParams3_1.nivelh = ListaParams3_0.nivelh$

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

Descriptor ::= Descriptor2[Exp]
Descriptor2$_1$.tsph = Descriptor2$_0$.tsph
Descriptor2$_1$.nivelh = Descriptor2$_0$.nivelh
Exp.tsph = Descriptor2$_0$.tsph
Exp.nivelh = Descriptor2$_0$.nivelh

Descriptor2 ::= ^Descriptor2
Descriptor2$_1$.tsph = Descriptor2$_0$.tsph
Descriptor2$_1$.nivelh = Descriptor2$_0$.nivelh

Exp ::= Exp OpRel ExpSum
Exp$_1$.tsph = Exp$_0$.tsph
ExpSum.tsph = Exp$_0$.Tsph
Exp.modo = valor

Exp ::= ExpSum
ExpSum.tsph = Exp.tsph
Exp.modo = ExpSum.modo

ExpSum ::= ExpSum OpAd ExpProd
ExpSum$_1$.tsph = ExpSum$_0$.tsph
ExpProd.tsph = ExpSum$_0$.tsph
ExpSum.modo = valor

ExpSum ::= ExpSum OR ExpProd
ExpSum$_1$.tsph = ExpSum$_0$.tsph
ExpProd.tsph = ExpSum$_0$.tsph
ExpSum.modo = valor

ExpSum ::= ExpProd
ExpProd.Tsph = ExpSum.Tsph
ExpSum.modo = ExpProd.modo

ExpProd ::= ExpProd OpProd ExpFact
$ExpProd_1$.tsph = $ExpProd_0$.tsph
ExpFact.tsph = $ExpProd_0$.tsph
ExpProd.modo = valor

ExpProd ::= ExpProd AND ExpFact
$ExpProd_1$.tsph = $ExpProd_0$.tsph
ExpFact.tsph = $ExpProd_0$.tsph
ExpProd.modo = valor

ExpProd ::= ExpFact
ExpFact.tsph = ExpProd.tsph
ExpProd.modo = ExpFact.modo

ExpFact ::= (Exp)
Exp.tsph = ExpFact.tsph
ExpFact.modo = Exp.modo

ExpFact:= OpAd numero
ExpFact.modo = valor

ExpFact := numero
ExpFact.modo = valor

ExpFact := True
ExpFact.modo = valor

ExpFact := False
ExpFact.modo = valor

ExpFact ::= Not ExpFact
$ExpFact_1$.Tsph = $ExpFact_0$.Tsph
$ExpFact_0$.modo = $ExpFact_1$.modo

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