Gramática Atribuida – Comprobación de Tipos

Nodo	Predicados	Reglas Semánticas
programa → *definiciones*:definicion*		
defCampo → *nombre*:String *tipo*:tipo		
defReturn → *tipo*:tipo	tipo ≠ IdentType	
cuerpo → *definicionesVariables*:defVariable* *sentencias*:sentencia*		
defVariable:definicion → *nombre*:String *tipo*:tipo *ambito*:String		
defFuncion:definicion → *nombre*:String *parametros*:defVariable* *retorno*:defReturn *cuerpo*:cuerpo		
defEstructura:definicion → *nombre*:String *campos*:defCampo*		
tipoEntero:tipo → λ		
tipoReal:tipo → λ		
tipoChar:tipo → λ		
tipoIdent:tipo → *nombre*:String		
tipoArray:tipo → *dimension*:literalEntero *tipo*:tipo		
asignacion:sentencia → *left*:expresion *right*:expresion	left.tipo ≠ IdentType left.modificable == TRUE left.tipo == right.tipo	
print:sentencia → *salida*:expresion	salida ≠ null salida.tipo ≠ tipoIdent	
printLN:sentencia → *salida*:expresion	salida.tipo ≠ tipoIdent	
printSP:sentencia → *salida*:expresion	salida ≠ null salida.tipo ≠ tipoIdent	

read:sentencia → *entrada*:expresion	entrada.modificable == TRUE entrada.tipo ≠ tipoIdent	
ifElse:sentencia → *condicion*:expresion *correcto*:sentencia* *incorrecto*:sentencia*	condición.tipo == tipoEntero	
while:sentencia → *condicion*:expresion *correcto*:sentencia*	condición.tipo == tipoEntero	
return:sentencia → *devolucion*:expresion	devolucion == null ⇔ definición.return == null devolucion ≠ null ⇔ definición.return ≠ null devolucion.tipo == definición.return.tipo	
invocacionProcedimiento:sentencia → *nombre*:String *argumentos*:expresion*	invocacionProcedimiento.definicion.parametros.size == argumentos.size invocacionProcedimiento.definicion .parametros$_i$.tipo == argumentos$_j$.tipo i==j	
literalEntero:expresion → *valor*:String		literalEntero.tipo = IntType literalEntero.modificable = FALSE
literalReal:expresion → *valor*:String		literalReal.tipo = RealType literalReal.modificable = FALSE
literalChar:expresion → *valor*:String		literalChar.tipo = CharType literalChar.modificable = FALSE
variable:expresion → *nombre*:String		variable.tipo = variable.definicion.tipo variable.modificable = TRUE
expresionAritmetica:expresion → *left*:expresion *operador*:String *right*:expresion	left.tipo == right.tipo	expresionAritmetica.modificable = FALSE expresionAritmetica.tipo = left.tipo = right.tipo
expresionComparacion:expresion → *left*:expresion *operador*:String *right*:expresion	left.tipo == right.tipo operador ∈ (&&, ||) ⇒ left.tipo = tipoEntero	expresionComparacion.modificable = FALSE expresionComparacion.tipo = tipoEntero

negacion:expresion → *expresion*:expresion	expresion.tipo == tipoEntero	negacion.modificable = FALSE negacion.tipo = tipoEntero
invocacionFuncion:expresion → *nombre*:String *argumentos*:expresion*	invocacionFuncion.definicion.parametros.size == argumentos.size invocacionFuncion.definicion.parametros$_i$.tipo == argumentos$_j$.tipo i==j	invocacionFuncion.tipo = invocacionFuncion.definicion.return.tipo invocacionFuncion.modificable = FALSE
cast:expresion → *tipo*:tipo *expresion*:expresion	tipo ≠ tipoIdent expresion.tipo ≠ tipoIdent tipo ≠ expresion.tipo	cast.tipo = tipo cast.modificable = FALSE
accesoArray:expresion → *identificador*:expresion *posicion*:expresion	posición.tipo == tipoIdent identificador.tipo == tipoArray	accesoArray.tipo = identificador.tipo.tipo accesoArray.modificable = TRUE
accesoCampo:expresion → *expresion*:expresion *campo*:String	expresion.tipo == tipoIdent estructuras[expresion.nombre].campos[nombre] ≠null	accesoCampos.tipo = estructuras[expresion.nombre].campos[nombre].tipo accesoCampos.modificable = TRUE
menosUnario:expresion → *expresion*:expresion		menosUnario.tipo = expresion.tipo menosUnario.modificable = expresion.modificable

Recordatorio de operadores (para cortar y pegar): ⇒ ⇔ ≠ ∅ ∈ ∉ ∪ ∩ ⊂ ⊄ ∑ ∃ ∀

Atributos

Categoría Sintáctica	Nombre del atributo	Tipo Java	Heredado/Sintetizado	Descripción
expresion	tipo	Tipo	Sintetizado	
expresion	modificable	boolean	Sintetizado	