

Gramática Atribuida

Símbolo	Predicados	Reglas semánticas
Programa	FOR definicion : definiciones SI definicion.nombre == 'main' RETURN TRUE FIN FOR RETURN FALSE	FOR definicion : definiciones SI definicion ES definicionVariable ENTONCES definicion.parametro = false
DefinicionVariable	SI (esParametro) tipo.primitivo	
DefinicionFuncion	retorno.primitivo	FOR parametro : parametros: parametro.esParametro=true FOR variable : variables: variable.esParametro = false FOR sentencia : sentencias: sentencia.definicionFuncion = this
DefinicionStruct		
AccesoArray	izq.tipo == TipoArray der.tipo == TipoEntero	tipo= izquierda.tipo.tipo lvalue = true
AccesoCampo	izq.tipo == DefinicionStruct der.tipo == Variable	tipo= campo.tipo lvalue = true
Aritmética	izq.tipo == der.tipo izq.tipo == (TipoInt O TipoReal) der.tipo == (TipoInt O TipoReal)	tipo = izq.tipo (o der.tipo) lvalue = false
Cast	tipo <> expresion.tipo tipo == (TipoInt O TipoReal O TipoChar) expresión.tipo == (TipoInt O TipoReal O TipoChar)	tipo = tipoCast lvalue = false
Comparacion	izq.tipo == der.tipo izq.tipo == (TipoInt O TipoReal O TipoChar) der.tipo == (TipoInt O	tipo = TipoEntero lvalue = false

	TipoReal O TipoChar)	
InvocacionFuncion	parametros.size == definicion.parametros.size PARA i=0 MIENTRAS i< parametros.size: SI parametros[i].tipo != definición.parametros[i].tipo RETURN FALSE FIN PARA RETURN TRUE node.tipo != null	tipo = definicionFuncion.retorno lvalue = false
LiteralCaracter		tipo = TipoChar lvalue = false
LiteralEntero		tipo = TipoInt lvalue = false
LiteralReal		tipo = TipoReal lvalue = false
Logica	op1.tipo = TipoInt op2.tipo = TipoInt	tipo = op1.tipo lvalue = false
Negacion		tipo = expr.tipo lvalue = false
Variable		tipo = definicion.tipo lvalue = true

Asignacion	izquierda.lvalue izquierda.tipo.primitivo izquierda.tipo == derecha.tipo	
IF	condicion.tipo == TipoInt	FOR sentencia : sentenciasIF: sentencia.definicionFuncion = definicionFuncion FOR sentencia:sentenciasElse: sentencia.definicionFuncion = definicionFuncion
InvocacionProc...	parametros.size == definicion.parametros.size PARA i=0 MIENTRAS i< parametros.size: SI parametros[i].tipo != definición.parametros[i].tipo RETURN FALSE	

	FIN PARA RETURN TRUE definicion.retorno == null /*Comprueba que la función que es llamada es realmente un procedimiento y no devuelve nada. Se evitan residuos en la pila.*/	
Print	expresion.tipo.primitivo	
Read	expresion.tipo.primitivo expresion.lvalue	
Return	SI expresion != NULL ENTONCES definicionFuncion.retorno == expresión SI expresion == NULL ENTONCES node.definicionFuncion.retorn o = null	
While	condicion.tipo == TipoInt	FOR sentencia : sentencias: sentencia.definicionFuncion = definicionFuncion

TipoArray	dimension > 0	primitive = false
TipoChar		primitive = true
TipoEntero		primitive = true
TipoReal		primitive = true
DefinicionStruct (como Tipo)		primitive = false

Nodo/Categoría	Atributo	Dominio (Tipo)	Heredado/Sintetizado
Tipo	primitivo	boolean	Sintetizado
DefinicionVariable	esParametro	boolean	Heredado
Expresion	tipo	Tipo	Sintetizado
Expresion	lvalue	Boolean	Sintetizado
Sentencia	definicionFuncion	DefinicionFuncion	Heredado