

Attribute Grammar

Símbolo	Predicados	Reglas Semánticas
programa → <i>definiciones</i> :defVariable* <i>sentencias</i> :sentencia*		
defVariable → <i>tipo</i> :tipo <i>nombre</i> :String		
intType :tipo → λ		
realType :tipo → λ		
print :sentencia → <i>expresion</i> :expresion		
asigna :sentencia → <i>left</i> :expresion <i>right</i> :expresion	mismoTipo(left.tipo, right.tipo) ¹ left.modificable	
exprAritmetica :expresion → <i>left</i> :expresion <i>operador</i> :String <i>right</i> :expresion	mismoTipo(left.tipo, right.tipo)	exprAritmetica.tipo = left.tipo exprAritmetica.modificable = false
variable :expresion → <i>nombre</i> :String		variable.tipo = definicion.tipo variable.modificable = true
literalInt :expresion → <i>valor</i> :String		literalInt.tipo = intType literalInt.modificable = false
literalReal :expresion → <i>valor</i> :String		literalReal.tipo = realType literalReal.modificable = false

Tabla de Atributos:

Categoría	Nombre	Tipo Java	H/S	Descripción
expresion	tipo	Tipo	Sintetizado	Tipo de la expresión (operaciones que admite)
expresion	modificable	boolean	Sintetizado	Indica si la expresión puede aparecer a la izquierda de una asignación

Funciones auxiliares:

mismoTipo(*tipoA*, *tipoB*) { tipoA == tipoB }

¹ La función auxiliar *mismoTipo* realmente no es necesaria y hubiera sido suficiente con poner directamente como predicado "*left.tipo == right.tipo*" en lugar del uso de la función. Sin embargo, se añade para posteriormente tener un ejemplo de cómo implementar funciones auxiliares como las que seguramente se presentarán en la práctica del alumno (*tipoMayor*, *primitivo*, *convertible*, etc.)