

# Attribute Grammar

Símbolo	Predicados	Reglas Semánticas
<b>program</b> → <i>definitions</i> :varDefinition* <i>sentences</i> :sentence*		
<b>varDefinition</b> → <i>type</i> :type <i>name</i> :String		
<b>intType</b> :type → λ		
<b>realType</b> :type → λ		
<b>print</b> :sentence → <i>expression</i> :expression		
<b>assignment</b> :sentence → <i>left</i> :expresión <i>right</i> :expression	mismoTipo(left.type, right.type) <sup>1</sup> left.modificable	
<b>arithmeticExpression</b> :expression → <i>left</i> :expresión <i>operator</i> :String <i>right</i> :expression	mismoTipo(left.type, right.type)	arithmeticExpression.type = left.type arithmeticExpression.modificable = false
<b>variable</b> :expression → <i>name</i> :String		variable.type = variable.definition.type variable.modificable = true
<b>intConstant</b> :expression → <i>valor</i> :String		intConstant.type = intType intConstant.modificable = false
<b>realConstant</b> :expression → <i>valor</i> :String		realConstant.type = realType realConstant.modificable = false

Tabla de Atributos

Nodo/Categoría	Nombre	Tipo Java	H/S	Descripción
expresion	type	Type	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 }

<sup>1</sup> La función auxiliar *mismoTipo* realmente no es necesaria y hubiera sido suficiente con poner directamente como predicado “*left.tipo == right.tipo*” en lugar de usar dicha 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.)