

Attribute Grammar

Nodo	Predicados	Reglas Semánticas
programa → <i>definiciones</i> :defVariable* <i>sentencias</i> :sentencia*		
defVariable → <i>tipo</i> :tipo <i>nombre</i> :String	variables[nombre] == ∅	variables[nombre] = defVariable
intType :tipo → λ		
realType :tipo → λ		
print :sentencia → <i>expresion</i> :expresion		
asigna :sentencia → <i>left</i> :expresion <i>right</i> :expresion		
exprAritmetica :expresion → <i>left</i> :expresion <i>operador</i> :String <i>right</i> :expresion		
variable :expresion → <i>nombre</i> :String	variables[nombre] ≠ ∅	variable.definicion = variables[nombre]
literalInt :expresion → <i>valor</i> :String		
literalReal :expresion → <i>valor</i> :String		

Atributos

Categoría Sintáctica	Nombre del atributo	Tipo Java	Heredado/Sintetizado	Descripción
variable	definicion	DefVariable	Sintetizado	Nodo donde está definido la variable

Conjuntos Auxiliares

variables	Map<String, DefVariable>
------------------	--------------------------

Recorrido de Gramáticas S-Atribuidas

Características de una Gramática S-Atribuida

- Puede implementarse con un solo recorrido del árbol
- La implementación de *todo* método *visit* debe seguir el patrón:

```

visit(Node nodoPadre) {
    Para cada hijoi de nodoPadre { // Da igual el orden de visita de los hijos
        Visitar(hijoi) // Al volver, el hijo tendrá ya asignados sus atributos
    }
    B(p): Comprobar predicados
    R(p): Asignar valor a los atributos de nodoPadre
}
    
```