

1 Code Specification

Nodo	Predicados	Reglas Semánticas
program → <i>ast</i> :AST*		dirAcumu = 0 Ast.forEach(<i>ast</i> -> if <i>ast</i> == defvar { <i>ast</i> .address = dirAcumu; dirAcumu += <i>ast</i> .type.size })
func :def → <i>name</i> :String <i>parameter</i> :parameter* <i>retorno</i> :type <i>defvar</i> :defVar* <i>sentence</i> :sentence*		parameter _i .address = 4 + \sum parameter _i .type.size 0 ≤ j < i defVar _i .address = \sum - defVar _i .type.size * defVar _i .name.size 0 ≤ j < i
defVar :def → <i>name</i> :String <i>type</i> :type		
parameter :def → <i>name</i> :String <i>type</i> :type		
defStruct :def → <i>name</i> :String <i>parameter</i> :parameter*		Parameter(i).address = \sum parameter.type.size 0 ≤ j < i
intType :type → λ		IntType.size = 2
realType :type → λ		realType.size = 4
charType :type → λ		CharType.size = 1
arrayType :type → <i>index</i> :int <i>type</i> :type		ArrayType.size = type.size * index
structType :type → <i>name</i> :String		StructType.size = \sum getStruct.param.size
voidType :type → λ		
print :sentence → <i>string</i> :String <i>expr</i> :expr		
read :sentence → <i>expr</i> :expr		
assignment :sentence → <i>left</i> :expr <i>right</i> :expr		
ifSentence :sentence → <i>condition</i> :expr <i>iftrue</i> :sentence*		
ifElseSentence :sentence → <i>condition</i> :expr <i>iftrue</i> :sentence* <i>else1</i> :sentence*		
whileSentence :sentence → <i>condition</i> :expr <i>sentence</i> :sentence*		
returnNode :sentence → <i>expr</i> :expr		
funcCall :sentence → <i>name</i> :String <i>args</i> :expr		

exprAritmetica: <i>expr</i> → <i>left:expr op:String right:expr</i>		
exprLogica: <i>expr</i> → <i>left:expr op:String right:expr</i>		
exprLogicaNe: <i>expr</i> → <i>expr:expr</i>		
acces: <i>expr</i> → <i>left:expr right:String</i>		
arrayAcces: <i>expr</i> → <i>left:expr right:expr</i>		
cast: <i>expr</i> → <i>typeToConvert:type expr:expr</i>		
litEnt: <i>expr</i> → <i>string:String</i>		
litReal: <i>expr</i> → <i>string:String</i>		
litChar: <i>expr</i> → <i>string:String</i>		
variable: <i>expr</i> → <i>string:String</i>		
methodCallExpr: <i>expr</i> → <i>name:String args:expr*</i>		

1.1 Atributos

Nodo/Categoría Sintáctica	Nombre del atributo	Tipo Java	Heredado/Sintetizado	Descripción
DefVar	Address	int	Heredado	
Type	size	int	Sintetizado	

1.2 Estructuras de Datos Auxiliares

Nombre	Tipo Java	Descripción

1.3 Funciones Auxiliares