

Definición formal de la traducción

Atributos semánticos

Producción	Atributos Sintetizados	Atributos Heredados
<i>Prog</i>	cod	
<i>Ident</i>		
<i>Iden</i>		
<i>Bloque</i>	cod	
<i>Tvar</i>	cod	
<i>Tvar2</i>		
<i>Tipo</i>		
<i>TBloque</i>	cod	
<i>TBloque2</i>	cod	
<i>TRead</i>	cod	
<i>TWrite</i>	cod	
<i>Text</i>	cod	
<i>TAsig</i>	cod	
<i>Exp</i>	cod	
<i>ExpSimple</i>	cod	
<i>Term</i>	cod	
<i>Fact</i>	cod	
<i>OpMul</i>	op	
<i>OpAd</i>	op	
<i>OpUn</i>	op	
<i>Comp</i>	op	

Gramática de atributos

$Prog ::= \text{program Ident PYCOMA Bloque PUNTO}$
 $\text{Prog.cod} = \text{Bloque.cod} \parallel \text{stop}$

$\text{Ident} ::= \text{id PA Iden PC}$

$\text{Iden} ::= \text{id}$

$\text{Iden} ::= \text{Iden COMA id}$

$\text{Bloque} ::= \text{TBloque}$
 $\text{Bloque.cod} = \text{TBloque.cod}$

$\text{Bloque} ::= \text{Tvar TBloque}$
 $\text{Bloque.cod} = \text{TBloque.cod}$

$\text{Tvar} ::= \text{var Tvar2}$
 $\text{Tvar.cod} = \text{Tvar2.cod}$

$\text{Tvar2} ::= \text{id 2PUNTOS Tipo PYCOMA}$

$\text{Tvar2} ::= \text{id 2PUNTOS Tipo PYCOMA Tvar2}$

$\text{Tipo} ::= \text{integer}$

$\text{Tipo} ::= \text{boolean}$

$\text{TBloque} ::= \text{begin TBloque2 end}$
 $\text{TBloque.cod} = \text{TBloque2.cod}$

$\text{TBloque2} ::= \lambda$

$\text{TBloque2} ::= \text{TAsig TBloque2}$
 $\text{TBloque2}_0.\text{cod} = \text{TAsig.cod} \parallel \text{TBloque2}_1.\text{cod}$

TBloque2 ::= TRead TBloque2
 TBloque2₀.cod = TRead.cod || TBloque2₁.cod

TBloque2 ::= TWrite TBloque2
 TBloque2₀.cod = TWrite.cod || TBloque2₁.cod

TRead ::= read TA id TC PYCOMA
 TRead.cod = lee || desapila_dir(dameDir(TRead.tsh,id.lex))

TWrite ::= write TA Text TC PYCOMA
 TWrite.cod = Text.cod

Text ::= texto
 Text.cod = apila(texto.lex) || escribe

Text ::= id
 Text.cod = apila(dameValor(Text.tsh,id.lex)) || escribe

TAsig ::= id ASIG Exp
 TAsig.cod = Exp.cod || desapila_dir(dameDir(TAsig.tsh,id.lex))

Exp ::= ExpSimple
 Exp.cod = ExpSimple.cod

Exp ::= ExpSimple Comp ExpSimple
 Exp.cod = ExpSimple₀.cod || ExpSimple₁.cod || Comp.op

ExpSimple ::= ExpSimple OpAd Term
 ExpSimple₀.cod = ExpSimple₁.cod || Term.cod || OpAd.op

ExpSimple ::= Term
 ExpSimple.cod = Term.cod

Term ::= Term OpMul Fact
 Term₀.cod = Term₁.cod || Fact.cod || OpMul.op

Term ::= Fact
 Term.cod = Fact.cod

Fact ::= numero
 Fact.cod = apila(valorDe(numero))

Fact ::= true
 ***Fact.cod* = apila(true)**

Fact ::= false
 ***Fact.cod* = apila(false)**

Fact ::= id
 ***Fact.cod* = apila_dir(dameDir(*Fact.tsh*,id.lex))**

Fact ::= *OpUn Fact*
 Fact₀.cod* = *Fact₁.cod* || *OpUn.op

Fact ::= (*Exp*)
 Fact.cod* = *Exp.cod

OpAd ::= +
 ***OpAd.op* = suma**

OpAd ::= -
 ***OpAd.op* = resta**

OpAd ::= or
 ***OpAd.op* = or**

OpMul ::= *
 ***OpMul.op* = multiplica**

OpMul ::= /
 ***OpMul.op* = divide**

OpMul ::= and
 ***OpMul.op* = and**

OpUn ::= +
 ***OpUn.op* = positivo**

OpUn ::= -
 ***OpUn.op* = negativo**

OpUn ::= not
 ***OpUn.op* = not**

Comp ::= <=
 ***Comp.op* = menor_igual**

Comp ::= >=
 Comp.op = mayor_igual

Comp ::= <
 Comp.op = menor

Comp ::= >
 Comp.op = mayor

Comp ::= =
 Comp.op = igual

Comp ::= ≠
 Comp.op = distinto