

Práctica
Procesadores de Lenguajes

David Antuña Rodríguez
Javier Carrión García

Contenidos

1	Fase 1	1
1.1	Clases léxicas	1
1.2	Especificación formal	3
1.3	Diagrama de transiciones	4
2	Fase 2	5
2.1	Gramática incontextual	5
2.2	Gramática transformada	6
2.3	Primeros, siguientes y directores	7
3	Fase 4	9
3.1	Funciones constructoras	9
3.2	Diagrama de clases	12
3.3	Gramática de atributos	13
3.4	Acondicionamiento para implementación descendente	15

1 Fase 1

1.1 Clases léxicas

La descripción de las clases léxicas identificadas se hará de manera informal, en lenguaje natural.

- **SPROG**

Es el separador && que indica el fin de la sección de declaraciones y el comienzo de la de instrucciones.

- **LREAL**

Empiezan con un signo (+ o -) opcional, a continuación aparecen uno o más dígitos cualesquiera. Seguida de esta parte puede aparecer una decimal que consta de un punto seguido de uno o más dígitos cualesquiera. Por último, tiene una E o e seguida de un signo (+ o -), opcional, y de uno o más dígitos cualesquiera.

- **ID**

Comienza por una letra cualquiera y la sigue una secuencia de cero o más letras, dígitos o subrayado(_).

- **BOOL**

Es una palabra reservada que se conforma por las letras minúsculas: b, o, o, l. En ese orden.

- **NUM**

Palabra reservada formada por las letras minúsculas: n, u, m. En ese orden.

- **TRUE**

Es una palabra reservada compuesta por las letras minúsculas: t, r, u, e. En ese orden.

- **FALSE**

Palabra reservada que contiene las siguientes letras minúsculas: f, a, l, s, e. En ese orden.

- **PLUS**

Representa una suma, \+.

- **MINUS**

Representa una resta, \-.

- **MUL**

Representa la multiplicación, *.

- **DIV**
Representa la división, $/$.
- **IS**
Representación de la asignación, $=$.
- **EQ**
Representa una comparación, $==$.
- **GT**
Representa el mayor que, $>$.
- **GEQ**
Representa el mayor o igual que, $>=$.
- **LT**
Representa el menor que, $<$.
- **LEQ**
Representa el menor o igual que, $<=$.
- **NEQ**
Representa una desigualdad, $!=$.
- **AND**
Representa el operador lógico and.
- **OR**
Representa el operador lógico or.
- **NOT**
Representa el operador lógico not.
- **POP**
Representa un paréntesis de apertura, $($.
- **PCL**
Representa un paréntesis de cierre, $)$.
- **EOL**
Representa el punto y coma como separador especial, $;$.

1.2 Especificación formal

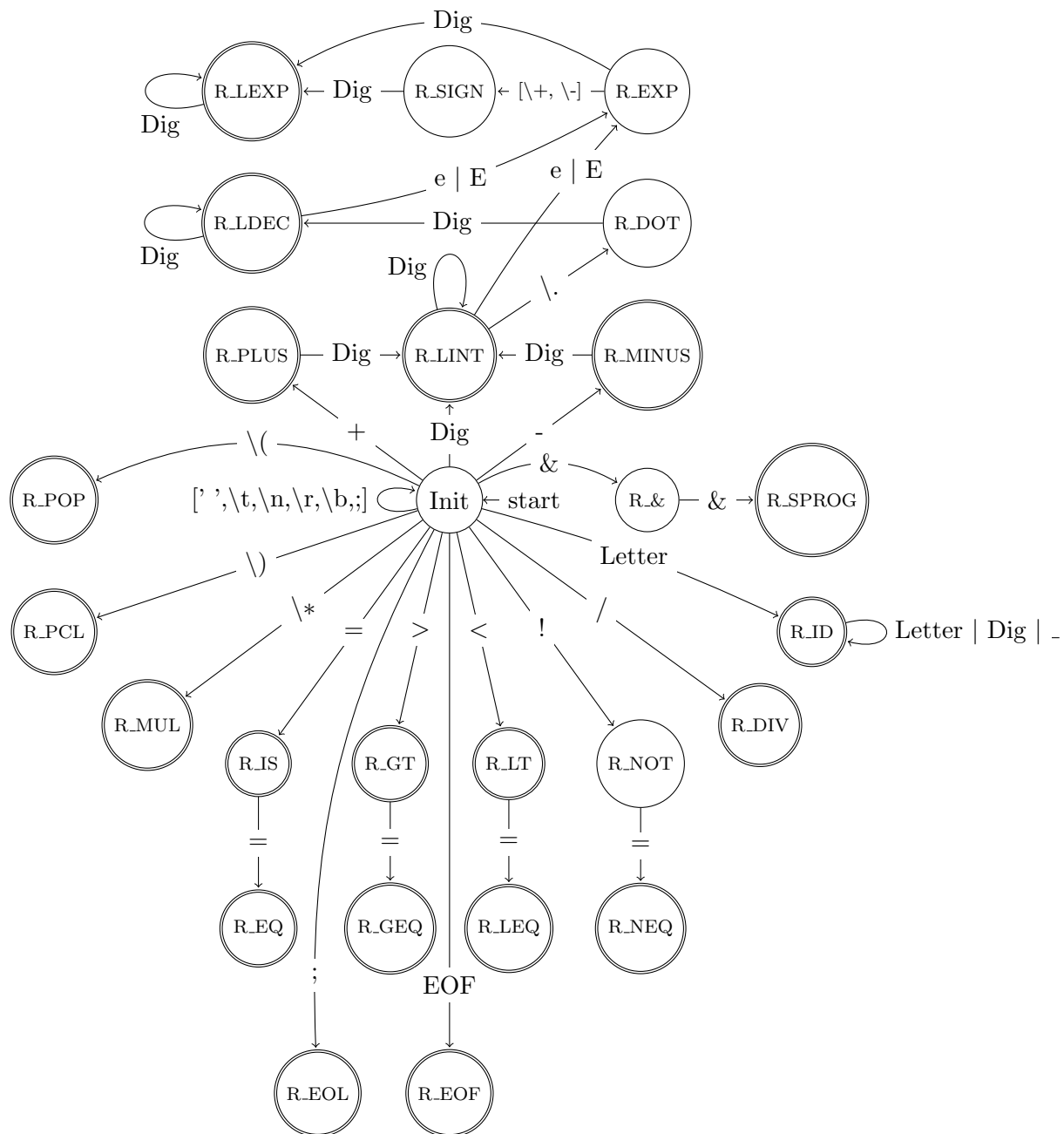
Vamos a utilizar DR_s para dar una descripción formal del lenguaje que conforman las clases léxicas del apartado 2.1.

- (*) $SPROG \equiv \&\&$
- (*) $LREAL \equiv \underline{L}ENT \ \underline{P}DEC? \ \underline{P}EXP?$
 $PDEC \equiv \backslash. \ \underline{Dig}^* \ \underline{Dig}$
 $PEXP \equiv (E \mid e) \ \underline{L}ENT$
 $LENT \equiv \underline{Sign}? \ \underline{Dig}^* \ \underline{Dig}$
 $Sign \equiv [\backslash+, \backslash-]$
 $Dig \equiv [0-9]$
- (*) $ID \equiv \underline{Letter} (\underline{Letter} \mid \underline{Dig} \mid -)^*$
 $Letter \equiv [a-z, A-Z]$
 $Dig \equiv [0-9]$
- (*) $BOOL \equiv b \ o \ o \ l$
- (*) $NUM \equiv n \ u \ m$
- (*) $TRUE \equiv t \ r \ u \ e$
- (*) $FALSE \equiv f \ a \ l \ s \ e$
- (*) $PLUS \equiv \backslash+$
- (*) $MINUS \equiv \backslash-$
- (*) $MUL \equiv \backslash*$
- (*) $DIV \equiv /$
- (*) $IS \equiv =$
- (*) $EQ \equiv ==$
- (*) $GT \equiv >$
- (*) $GEQ \equiv >=$
- (*) $LT \equiv <$
- (*) $LEQ \equiv <=$
- (*) $NEQ \equiv !=$
- (*) $AND \equiv a \ n \ d$
- (*) $OR \equiv o \ r$
- (*) $NOT \equiv n \ o \ t$
- (*) $POP \equiv \backslash($
- (*) $PCL \equiv \backslash)$

$$(*) \text{ EOL} \equiv ;$$

[I] SEP \equiv [' ', \t, \n, \r, \b]

1.3 Diagrama de transiciones



2 Fase 2

2.1 Gramática incontextual

La especificación sintáctica utilizara la siguiente tabla de operadores.

Operador	Prioridad	Aridad	Tipo	Asociatividad
+, −	0	2	infijo	izq.
and	1	2	infijo	der.
or	1	2	infijo	no
<, >, ==, !=, <=, >=	2	2	infijo	no
*, /	3	2	infijo	izq.
−	4	1	prefijo	si
not	4	1	prefijo	no

Con los operadores definidos procedemos a realizar la especificación.

$S \rightarrow \text{Prog EOF}$

$\text{Prog} \rightarrow \text{LDec SPROG LIns}$

$\text{LDec} \rightarrow \text{LDec EOL Dec}$

$\text{LDec} \rightarrow \text{Dec}$

$\text{LIns} \rightarrow \text{LIns EOL Ins}$

$\text{LIns} \rightarrow \text{Ins}$

$\text{Dec} \rightarrow \text{NUM ID}$

$\text{Dec} \rightarrow \text{BOOL ID}$

$\text{Ins} \rightarrow \text{ID IS Exp0}$

$\text{Exp0} \rightarrow \text{Exp0 Op0 Exp1}$

$\text{Exp0} \rightarrow \text{Exp1}$

$\text{Exp1} \rightarrow \text{Exp2 AND Exp1}$

$\text{Exp1} \rightarrow \text{Exp2 OR Exp2}$

$\text{Exp1} \rightarrow \text{Exp2}$

$\text{Exp2} \rightarrow \text{Exp3 Op2 Exp3}$

$\text{Exp2} \rightarrow \text{Exp3}$

$\text{Exp3} \rightarrow \text{Exp3 Op3 Exp4}$

$\text{Exp3} \rightarrow \text{Exp4}$

$\text{Exp4} \rightarrow \text{MINUS Exp4}$

$\text{Exp4} \rightarrow \text{NOT Exp5}$

$\text{Exp4} \rightarrow \text{Exp5}$

$\text{Exp5} \rightarrow \text{LREAL}$

$\text{Exp5} \rightarrow \text{TRUE}$

$\text{Exp5} \rightarrow \text{FALSE}$

$\text{Exp5} \rightarrow \text{POP Exp0 PCL}$

$\text{Op0} \rightarrow \text{PLUS}$

$\text{Op0} \rightarrow \text{MINUS}$

$\text{Op2} \rightarrow \text{EQ}$

$\text{Op2} \rightarrow \text{GT}$

$\text{Op2} \rightarrow \text{GEQ}$

$\text{Op2} \rightarrow \text{LT}$

$\text{Op2} \rightarrow \text{LEQ}$

$\text{Op2} \rightarrow \text{NEQ}$

$\text{Op3} \rightarrow \text{MUL}$

$\text{Op3} \rightarrow \text{DIV}$

2.2 Gramática transformada

Vamos a transformar la gramática de la sección anterior para obtener una gramática LL(1) equivalente.

$S \rightarrow \text{Prog } \underline{\text{EOF}}$
 $\text{Prog} \rightarrow \text{SDec } \underline{\text{SPROG}} \text{ SIns}$
 $\text{SDec} \rightarrow \text{Dec LDec}$
 $\text{LDec} \rightarrow \underline{\text{EOL}} \text{ Dec LDec}$
 $\text{LDec} \rightarrow \varepsilon$
 $\text{SIns} \rightarrow \text{Ins LIns}$
 $\text{LIns} \rightarrow \underline{\text{EOL}} \text{ Ins LIns}$
 $\text{LIns} \rightarrow \varepsilon$
 $\text{Dec} \rightarrow \underline{\text{NUM}} \underline{\text{ID}}$
 $\text{Dec} \rightarrow \underline{\text{BOOL}} \underline{\text{ID}}$
 $\text{Ins} \rightarrow \underline{\text{ID}} \underline{\text{IS}} \text{ Exp0}$

$\text{Exp0} \rightarrow \text{Exp1 R0}$
 $\text{R0} \rightarrow \text{Op0 Exp1 R0}$
 $\text{R0} \rightarrow \varepsilon$
 $\text{Exp1} \rightarrow \text{Exp2 R1}$
 $\text{R1} \rightarrow \underline{\text{AND}} \text{ Exp2 R1}$
 $\text{R1} \rightarrow \underline{\text{OR}} \text{ Exp2}$
 $\text{R1} \rightarrow \varepsilon$
 $\text{Exp2} \rightarrow \text{Exp3 R2}$
 $\text{R2} \rightarrow \text{Op2 Exp3 R2}$
 $\text{R2} \rightarrow \varepsilon$
 $\text{Exp3} \rightarrow \text{Exp4 R3}$
 $\text{R3} \rightarrow \text{Op3 Exp4 R3}$
 $\text{R3} \rightarrow \varepsilon$
 $\text{Exp4} \rightarrow \underline{\text{MINUS}} \text{ Exp4}$
 $\text{Exp4} \rightarrow \underline{\text{NOT}} \text{ Exp5}$
 $\text{Exp4} \rightarrow \text{Exp5}$
 $\text{Exp5} \rightarrow \underline{\text{LREAL}}$
 $\text{Exp5} \rightarrow \underline{\text{TRUE}}$
 $\text{Exp5} \rightarrow \underline{\text{FALSE}}$
 $\text{Exp5} \rightarrow \underline{\text{POP}} \text{ Exp0 } \underline{\text{PCL}}$

$\text{Op0} \rightarrow \underline{\text{PLUS}}$
 $\text{Op0} \rightarrow \underline{\text{MINUS}}$
 $\text{Op2} \rightarrow \underline{\text{EQ}}$
 $\text{Op2} \rightarrow \underline{\text{GT}}$
 $\text{Op2} \rightarrow \underline{\text{GEQ}}$
 $\text{Op2} \rightarrow \underline{\text{LT}}$
 $\text{Op2} \rightarrow \underline{\text{LEQ}}$
 $\text{Op2} \rightarrow \underline{\text{NEQ}}$
 $\text{Op3} \rightarrow \underline{\text{MUL}}$
 $\text{Op3} \rightarrow \underline{\text{DIV}}$

2.3 Primeros, siguientes y directores

Hemos utilizado la herramienta [The Context Free Grammar Checker](#) de la Universidad de Calgary para generar automáticamente los sets de símbolos. Para generarlos tan solo es necesario ir a la web y copiar el contenido del fichero transformed_grammar.md.

```
PRIM(S) = {NUM, BOOL}
PRIM(Prog) = {NUM, BOOL}
PRIM(SDec) = {NUM, BOOL}
PRIM(LDec) = {EOL, ε}
PRIM(SIns) = {ID}
PRIM(LIns) = {EOL, ε}
PRIM(Dec) = {NUM, BOOL}
PRIM(Ins) = {ID}
PRIM(R0) = {PLUS, MINUS, ε}
PRIM(Exp1) = {MINUS, NOT, LREAL, TRUE, FALSE, POP}
PRIM(R1) = {AND, OR, ε}
PRIM(Exp2) = {MINUS, NOT, LREAL, TRUE, FALSE, POP}
PRIM(R2) = {EQ, GT, GEQ, LT, LEQ, NEQ, ε}
PRIM(Exp3) = {MINUS, NOT, LREAL, TRUE, FALSE, POP}
PRIM(R3) = {MUL, DIV, ε}
PRIM(Exp4) = {MINUS, NOT, LREAL, TRUE, FALSE, POP}
PRIM(Exp5) = {LREAL, TRUE, FALSE, POP}
PRIM(Exp0) = {MINUS, NOT, LREAL, TRUE, FALSE, POP}
PRIM(Op0) = {PLUS, MINUS}
PRIM(Op2) = {EQ, GT, GEQ, LT, LEQ, NEQ}
PRIM(Op3) = {MUL, DIV}

SIG(S) = ∅
SIG(Prog) = {EOF}
SIG(SDec) = {SPROG}
SIG(LDec) = {SPROG}
SIG(SIns) = {EOF}
SIG(LIns) = {EOF}
SIG(Dec) = {EOL, SPROG}
SIG(Ins) = {EOL, EOF}
SIG(R0) = {PCL, EOL, EOF}
SIG(Exp1) = {PCL, PLUS, MINUS, EOL, EOF}
SIG(R1) = {PCL, PLUS, MINUS, EOL, EOF}
SIG(Exp2) = {PCL, AND, OR, PLUS, MINUS, EOL, EOF}
SIG(R2) = {PCL, AND, OR, PLUS, MINUS, EOL, EOF}
SIG(Exp3) = {PCL, EQ, GT, GEQ, LT, LEQ, NEQ, AND, OR, PLUS, MINUS,
             EOL, EOF}
SIG(R3) = {PCL, EQ, GT, GEQ, LT, LEQ, NEQ, AND, OR, PLUS, MINUS, EOL,
             EOF}
```

$SIG(Exp4) = \{PCL, MUL, DIV, EQ, GT, GEQ, LT, LEQ, NEQ, AND, OR, PLUS, MINUS, EOL, EOF\}$
 $SIG(Exp5) = \{PCL, MUL, DIV, EQ, GT, GEQ, LT, LEQ, NEQ, AND, OR, PLUS, MINUS, EOL, EOF\}$
 $SIG(Exp0) = \{PCL, EOL, EOF\}$
 $SIG(Op0) = \{MINUS, NOT, LREAL, TRUE, FALSE, POP\}$
 $SIG(Op2) = \{MINUS, NOT, LREAL, TRUE, FALSE, POP\}$
 $SIG(Op3) = \{MINUS, NOT, LREAL, TRUE, FALSE, POP\}$
 $DIR(S \rightarrow Prog \underline{EOF}) = \{NUM, BOOL\}$
 $DIR(Prog \rightarrow SDec \underline{SPROG} SIns) = \{NUM, BOOL\}$
 $DIR(SDec \rightarrow Dec LDec) = \{NUM, BOOL\}$
 $DIR(LDec \rightarrow \underline{EOL} Dec LDec) = \{EOL\}$
 $DIR(LDec \rightarrow \varepsilon) = \{SPROG\}$
 $DIR(SIns \rightarrow Ins LIns) = \{ID\}$
 $DIR(LIns \rightarrow \underline{EOL} Ins LIns) = \{EOL\}$
 $DIR(LIns \rightarrow \varepsilon) = \{EOF\}$
 $DIR(Dec \rightarrow \underline{NUM} ID) = \{NUM\}$
 $DIR(Dec \rightarrow \underline{BOOL} ID) = \{BOOL\}$
 $DIR(Ins \rightarrow ID \underline{EQ} Exp0) = \{ID\}$
 $DIR(Exp0 \rightarrow Exp1 R0) = \{MINUS, NOT, LREAL, TRUE, FALSE, POP\}$
 $DIR(R0 \rightarrow Op0 Exp1 R0) = \{PLUS, MINUS\}$
 $DIR(R0 \rightarrow \varepsilon) = \{PCL, EOL, EOF\}$
 $DIR(Exp1 \rightarrow Exp2 R1) = \{MINUS, NOT, LREAL, TRUE, FALSE, POP\}$
 $DIR(R1 \rightarrow \underline{AND} Exp2 R1) = \{AND\}$
 $DIR(R1 \rightarrow \underline{OR} Exp2) = \{OR\}$
 $DIR(R1 \rightarrow \varepsilon) = \{PCL, PLUS, MINUS, EOL, EOF\}$
 $DIR(Exp2 \rightarrow Exp3 R2) = \{MINUS, NOT, LREAL, TRUE, FALSE, POP\}$
 $DIR(R2 \rightarrow Op2 Exp3 R2) = \{EQ, GT, GEQ, LT, LEQ, NEQ\}$
 $DIR(R2 \rightarrow \varepsilon) = \{PCL, AND, OR, PLUS, MINUS, EOL, EOF\}$
 $DIR(Exp3 \rightarrow Exp4 R3) = \{MINUS, NOT, LREAL, TRUE, FALSE, POP\}$
 $DIR(R3 \rightarrow Op3 Exp4 R3) = \{MUL, DIV\}$
 $DIR(R3 \rightarrow \varepsilon) = \{PCL, EQ, GT, GEQ, LT, LEQ, NEQ, AND, OR, PLUS, MINUS, EOL, EOF\}$
 $DIR(Exp4 \rightarrow \underline{MINUS} Exp4) = \{MINUS\}$
 $DIR(Exp4 \rightarrow \underline{NOT} Exp5) = \{NOT\}$
 $DIR(Exp4 \rightarrow Exp5) = \{LREAL, TRUE, FALSE, POP\}$
 $DIR(Exp5 \rightarrow \underline{LREAL}) = \{LREAL\}$
 $DIR(Exp5 \rightarrow \underline{TRUE}) = \{TRUE\}$
 $DIR(Exp5 \rightarrow \underline{FALSE}) = \{FALSE\}$
 $DIR(Exp5 \rightarrow \underline{POP} Exp0 \underline{PCL}) = \{POP\}$
 $DIR(Op0 \rightarrow \underline{PLUS}) = \{PLUS\}$
 $DIR(Op0 \rightarrow \underline{MINUS}) = \{MINUS\}$
 $DIR(Op2 \rightarrow \underline{EQ}) = \{EQ\}$

$\text{DIR}(\text{Op2} \rightarrow \underline{\text{GT}}) = \{\text{GT}\}$
 $\text{DIR}(\text{Op2} \rightarrow \underline{\text{GEQ}}) = \{\text{GEQ}\}$
 $\text{DIR}(\text{Op2} \rightarrow \underline{\text{LT}}) = \{\text{LT}\}$
 $\text{DIR}(\text{Op2} \rightarrow \underline{\text{LEQ}}) = \{\text{LEQ}\}$
 $\text{DIR}(\text{Op2} \rightarrow \underline{\text{NEQ}}) = \{\text{NEQ}\}$
 $\text{DIR}(\text{Op3} \rightarrow \underline{\text{MUL}}) = \{\text{MUL}\}$
 $\text{DIR}(\text{Op3} \rightarrow \underline{\text{DIV}}) = \{\text{DIV}\}$

3 Fase 4

3.1 Funciones constructoras

El primer paso para poder generar las funciones constructoras es simplificar la gramática incontextual generada en la fase 2.

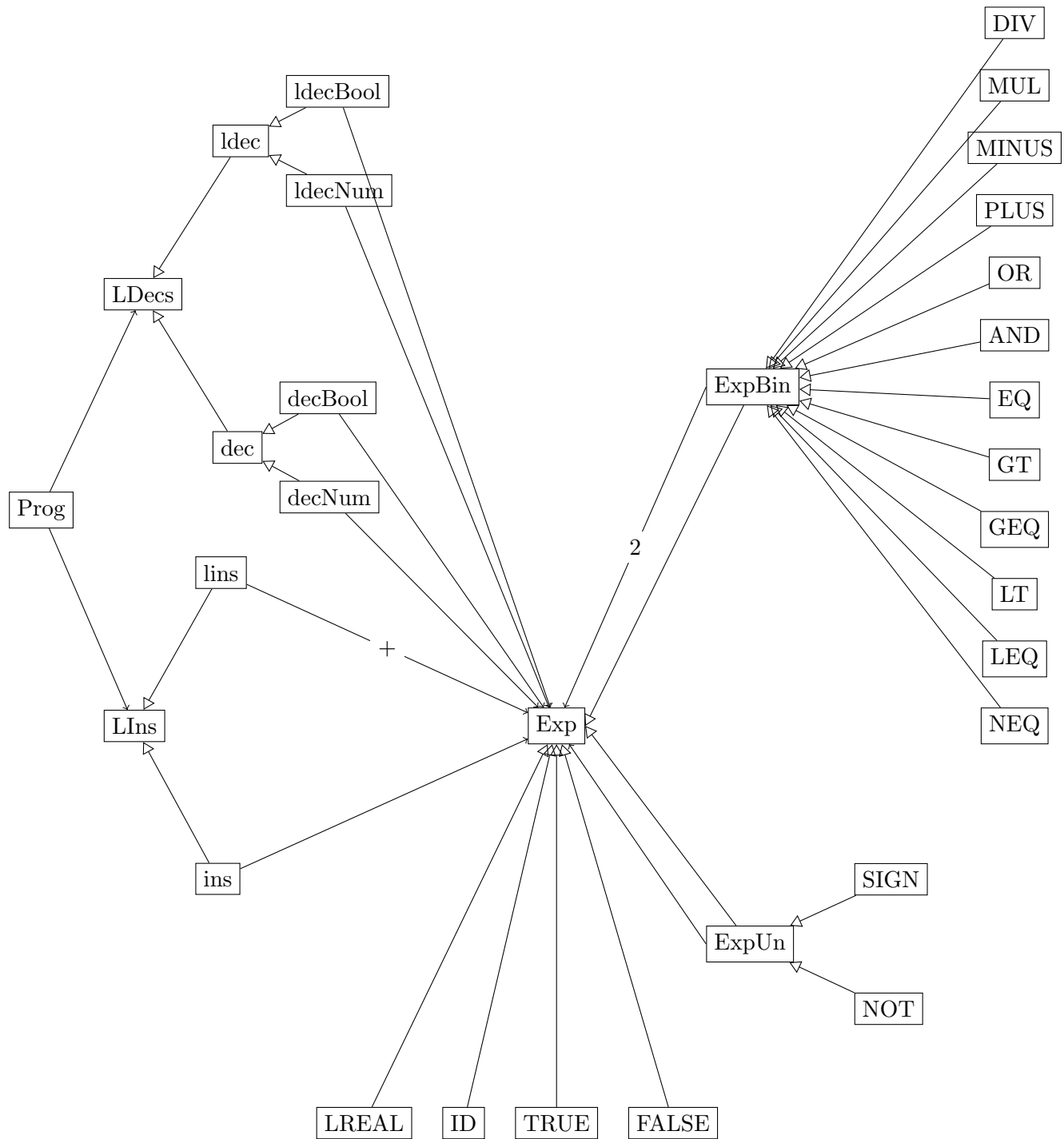
$\text{Prog} \rightarrow \text{LDec } \underline{\text{SPROG}} \text{ LIns}$
 $\text{LDec} \rightarrow \text{LDec } \underline{\text{EOL}} \text{ NUM } \underline{\text{ID}}$
 $\quad | \text{LDec } \underline{\text{EOL}} \text{ BOOL } \underline{\text{ID}}$
 $\quad | \underline{\text{NUM}} \underline{\text{ID}}$
 $\quad | \underline{\text{BOOL}} \underline{\text{ID}}$
 $\text{LIns} \rightarrow \text{LIns } \underline{\text{EOL}} \underline{\text{ID}} \text{ IS } \text{Exp}$
 $\quad | \underline{\text{ID}} \text{ IS } \text{Exp}$
 $\text{Exp} \rightarrow \text{Exp } \underline{\text{PLUS}} \text{ Exp}$
 $\quad | \text{Exp } \underline{\text{MINUS}} \text{ Exp}$
 $\quad | \text{Exp } \underline{\text{AND}} \text{ Exp}$
 $\quad | \text{Exp } \underline{\text{OR}} \text{ Exp}$
 $\quad | \text{Exp } \underline{\text{EQ}} \text{ Exp}$
 $\quad | \text{Exp } \underline{\text{GT}} \text{ Exp}$
 $\quad | \text{Exp } \underline{\text{GEQ}} \text{ Exp}$
 $\quad | \text{Exp } \underline{\text{LT}} \text{ Exp}$
 $\quad | \text{Exp } \underline{\text{LEQ}} \text{ Exp}$
 $\quad | \text{Exp } \underline{\text{NEQ}} \text{ Exp}$
 $\quad | \text{Exp } \underline{\text{MUL}} \text{ Exp}$
 $\quad | \text{Exp } \underline{\text{DIV}} \text{ Exp}$
 $\quad | \underline{\text{MINUS}} \text{ Exp}$
 $\quad | \underline{\text{NOT}} \text{ Exp}$
 $\quad | \underline{\text{LREAL}}$
 $\quad | \underline{\text{TRUE}}$
 $\quad | \underline{\text{FALSE}}$
 $\quad | \underline{\text{POP}} \text{ Exp } \underline{\text{PCL}}$

Por cada una de las reglas de esta gramática simplificada generaremos una función constructora.

- $\text{Prog} \rightarrow \text{LDec } \underline{\text{SPROG}} \text{ LIns}$
 $\text{prog}: \text{LDec} \times \text{LIns} \rightarrow \text{Prog}$
- $\text{LDec} \rightarrow \text{LDec } \underline{\text{EOL}} \underline{\text{NUM}} \underline{\text{ID}}$
 $\text{ldecNum}: \text{LDec} \times \text{String} \rightarrow \text{LDec}$
- $\text{LDec} \rightarrow \text{LDec } \underline{\text{EOL}} \underline{\text{BOOL}} \underline{\text{ID}}$
 $\text{ldecBool}: \text{LDec} \times \text{String} \rightarrow \text{LDec}$
- $\text{LDec} \rightarrow \underline{\text{NUM}} \underline{\text{ID}}$
 $\text{decNum}: \text{String} \rightarrow \text{LDec}$
- $\text{LDec} \rightarrow \underline{\text{BOOL}} \underline{\text{ID}}$
 $\text{decBool}: \text{String} \rightarrow \text{LDec}$
- $\text{LIns} \rightarrow \text{LIns } \underline{\text{EOL}} \underline{\text{ID}} \underline{\text{IS}} \text{ Exp}$
 $\text{lins}: \text{LIns} \times \text{String} \times \text{Exp} \rightarrow \text{LIns}$
- $\text{LIns} \rightarrow \underline{\text{ID}} \underline{\text{IS}} \text{ Exp}$
 $\text{ins}: \text{String} \times \text{Exp} \rightarrow \text{LIns}$
- $\text{Exp} \rightarrow \text{Exp } \underline{\text{PLUS}} \text{ Exp}$
 $\text{plus}: \text{Exp} \times \text{Exp} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \text{Exp } \underline{\text{MINUS}} \text{ Exp}$
 $\text{minus}: \text{Exp} \times \text{Exp} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \text{Exp } \underline{\text{AND}} \text{ Exp}$
 $\text{and}: \text{Exp} \times \text{Exp} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \text{Exp } \underline{\text{OR}} \text{ Exp}$
 $\text{or}: \text{Exp} \times \text{Exp} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \text{Exp } \underline{\text{EQ}} \text{ Exp}$
 $\text{eq}: \text{Exp} \times \text{Exp} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \text{Exp } \underline{\text{GT}} \text{ Exp}$
 $\text{gt}: \text{Exp} \times \text{Exp} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \text{Exp } \underline{\text{GEQ}} \text{ Exp}$
 $\text{geq}: \text{Exp} \times \text{Exp} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \text{Exp } \underline{\text{LT}} \text{ Exp}$
 $\text{lt}: \text{Exp} \times \text{Exp} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \text{Exp } \underline{\text{LEQ}} \text{ Exp}$
 $\text{leq}: \text{Exp} \times \text{Exp} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \text{Exp } \underline{\text{NEQ}} \text{ Exp}$
 $\text{neq}: \text{Exp} \times \text{Exp} \rightarrow \text{Exp}$

- $\text{Exp} \rightarrow \text{Exp } \underline{\text{MUL}} \text{ Exp}$
mul: $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \text{Exp } \underline{\text{DIV}} \text{ Exp}$
div: $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \underline{\text{MINUS}} \text{ Exp}$
sign: $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \underline{\text{NOT}} \text{ Exp}$
not: $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \underline{\text{LREAL}}$
num: $\text{String} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \underline{\text{TRUE}}$
true: $\text{String} \rightarrow \text{Exp}$
- $\text{Exp} \rightarrow \underline{\text{FALSE}}$
false: $\text{String} \rightarrow \text{Exp}$

3.2 Diagrama de clases



3.3 Gramática de atributos

$S \rightarrow \text{Prog } \underline{\text{EOF}}$
 $S.a = \text{Prog}.a$
 $\text{Prog} \rightarrow \text{LDec } \underline{\text{SPROG}} \text{ LIns}$
 $\text{Prog}.a = \text{prog}(\text{LDec}.a, \text{LIns}.a)$
 $\text{LDec} \rightarrow \text{LDec } \underline{\text{EOL}} \text{ Dec}$
 $\text{LDec}_0.a = \text{ldec}(\text{LDec}_1.a, \text{Dec}.a)$
 $\text{LDec} \rightarrow \text{Dec}$
 $\text{LDec}.a = \text{Dec}.a$
 $\text{LIns} \rightarrow \text{LIns } \underline{\text{EOL}} \text{ Ins}$
 $\text{LIns}_0.a = \text{lins}(\text{LIns}_1.a, \text{Ins}.id, \text{Ins}.exp)$
 $\text{LIns} \rightarrow \text{Ins}$
 $\text{LIns}.a = \text{ins}(\text{Ins}.id, \text{Ins}.exp)$
 $\text{Dec} \rightarrow \underline{\text{NUM}} \underline{\text{ID}}$
 $\text{Dec}.a = \text{decNum}(\text{ID}.lex)$
 $\text{Dec} \rightarrow \underline{\text{BOOL}} \underline{\text{ID}}$
 $\text{Dec}.a = \text{decBool}(\text{ID}.lex)$
 $\text{Ins} \rightarrow \underline{\text{ID}} \underline{\text{IS}} \text{ Exp0}$
 $\text{Ins}.id = \text{ID}.lex$
 $\text{Ins}.exp = \text{Exp0}.exp$
 $\text{Exp0} \rightarrow \text{Exp0 Op0 Exp1}$
 $\text{Exp0}_0.a = \text{mkexpb}(\text{Op0}.op, \text{Exp0}_1.a, \text{Exp1}.a)$
 $\text{Exp0} \rightarrow \text{Exp1}$
 $\text{Exp0}.a = \text{Exp1}.a$
 $\text{Exp1} \rightarrow \text{Exp2 } \underline{\text{AND}} \text{ Exp1}$
 $\text{Exp1}_0.a = \text{mkexpb}('and', \text{Exp2}.a, \text{Exp1}_1.a)$
 $\text{Exp1} \rightarrow \text{Exp2 } \underline{\text{OR}} \text{ Exp2}$
 $\text{Exp1}.a = \text{mkexpb}('or', \text{Exp2}_0.a, \text{Exp2}_1.a)$
 $\text{Exp1} \rightarrow \text{Exp2}$
 $\text{Exp1}.a = \text{Exp2}.a$
 $\text{Exp2} \rightarrow \text{Exp3 Op2 Exp3}$
 $\text{Exp2}.a = \text{mkexpb}(\text{Op2}.op, \text{Exp3}_0.a, \text{Exp3}_1.a)$
 $\text{Exp2} \rightarrow \text{Exp3}$
 $\text{Exp2}.a = \text{Exp3}.a$
 $\text{Exp3} \rightarrow \text{Exp3 Op3 Exp4}$
 $\text{Exp3}_0.a = \text{mkexpb}(\text{Op3}.a, \text{Exp3}_1.a, \text{Exp4}.a)$
 $\text{Exp3} \rightarrow \text{Exp4}$
 $\text{Exp3}.a = \text{Exp4}.a$
 $\text{Exp4} \rightarrow \underline{\text{MINUS}} \text{ Exp4}$
 $\text{Exp4}_0.a = \text{mkexpu}('-', \text{Exp4}_1.a)$
 $\text{Exp4} \rightarrow \underline{\text{NOT}} \text{ Exp5}$
 $\text{Exp4}.a = \text{mkexpu}('not', \text{Exp5}.a)$
 $\text{Exp4} \rightarrow \text{Exp5}$

```

    Exp4.a = Exp5.a
Exp5 → LREAL
    Exp5.a = num(LREAL.lex)
Exp5 → TRUE
    Exp5.a = true(TRUE.lex)
Exp5 → FALSE
    Exp5.a = false(FALSE.lex)
Exp5 → POP Exp0 PCL
    Exp5.a = Exp0.a
Op0 → PLUS
    Op0.op = '+'
Op0 → MINUS
    Op0.op = '-'
Op2 → EQ
    Op2.op = '==',
Op2 → GT
    Op2.op = '>',
Op2 → GEQ
    Op2.op = '>=',
Op2 → LT
    Op2.op = '<',
Op2 → LEQ
    Op2.op = '<=',
Op2 → NEQ
    Op2.op = '!=',
Op3 → MUL
    Op3.op = '*',
Op3 → DIV
    Op3.op = '/'

```

Las funciones mkexpb y mkexpu son las encargadas de construir las expresiones apropiadas para operadores binarios y unarios, respectivamente.

```

fun mkexpb (op, opd1, opd2) {
  switch(op) {
    '+': return plus(opd1, opd2)
    '-': return minus(opd1, opd2)
    '*': return mul(opd1, opd2)
    '/': return div(opd1, opd2)
    'and': return and(opd1, opd2)
    'or': return or(opd1, opd2)
    '==': return eq(opd1, opd2)
    '>': return gt(opd1, opd2)
    '>=': return geq(opd1, opd2)
  }
}

```



```

    '<': return lt(opd1, opd2)
    '<=': return leq(opd1, opd2)
    '!=': return neq(opd1, opd2)
  }
}
fun mkexpu (op, opd) {
  switch(op) {
    '-': return sign(opd)
    'not': return not(opd)
  }
}

```

3.4 Acondicionamiento para implementación descendente

```

S → Prog EOF
  S.a = Prog.a
Prog → SDec SPROG SIns
  Prog.a = prog(LDec.a, LIns.a)
SDec → Dec LDec
  LDec.ah = Dec.a
  SDec.a = LDec.a
LDec → EOL Dec LDec
  LDec1.a = ldec(LDec1.ah, Dec.a)
  LDec0.a = LDec1.a
LDec → ε
  LDec.a = LDec.ah
SIns → Ins LIns
  LIns.ah = ins(Ins.id, Ins.exp)
  SIns.a = LIns.a
LIns → EOL Ins LIns
  LIns1.a = lins(LIns1.ah, Ins.id, Ins.exp)
  LIns0.a = LIns1.a
LIns → ε
  LIns.a = LIns.ah
Dec → NUM ID
  Dec.a = decNum(ID.lex)
Dec → BOOL ID
  Dec.a = decBool(ID.lex)
Ins → ID IS Exp0
  Ins.id = ID.lex
  Ins.exp = Exp0.exp
Exp0 → Exp1 R0
  R0.ah = Exp1.a

```

$\text{Exp0.a} = \text{R0.a}$
 $\text{R0} \rightarrow \text{Op0 Exp1 R0}$
 $\text{R0}_1.\text{ah} = \text{mkexpb}(\text{Op0.op}, \text{R0}_0.\text{ah}, \text{Exp1.a})$
 $\text{R0}_0.\text{a} = \text{R0}_1.\text{a}$
 $\text{R0} \rightarrow \varepsilon$
 $\text{R0.a} = \text{R0.ah}$
 $\text{Exp1} \rightarrow \text{Exp2 R1}$
 $\text{R1.ah} = \text{Exp2.a}$
 $\text{Exp1.a} = \text{R1.a}$
 $\text{R1} \rightarrow \underline{\text{AND}} \text{Exp2 R1}$
 $\text{R1}_1.\text{ah} = \text{mkexpb}(\text{'and'}, \text{R1}_0.\text{ah}, \text{Exp2.a})$
 $\text{R1}_0.\text{a} = \text{R1}_1.\text{a}$
 $\text{R1} \rightarrow \underline{\text{OR}} \text{Exp2}$
 $\text{R1.a} = \text{mkexpb}(\text{'or'}, \text{R1.ah}, \text{Exp2.a})$
 $\text{R1} \rightarrow \varepsilon$
 $\text{R1.a} = \text{R1.ah}$
 $\text{Exp2} \rightarrow \text{Exp3 R2}$
 $\text{R2.ah} = \text{Exp3.a}$
 $\text{Exp2.a} = \text{R2.a}$
 $\text{R2} \rightarrow \text{Op2 Exp3 R2}$
 $\text{R2}_1.\text{ah} = \text{mkexpb}(\text{Op2.op}, \text{R2}_0.\text{ah}, \text{Exp3.a})$
 $\text{R2}_0.\text{a} = \text{R2}_1.\text{a}$
 $\text{R2} \rightarrow \varepsilon$
 $\text{R2.a} = \text{R2.ah}$
 $\text{Exp3} \rightarrow \text{Exp4 R3}$
 $\text{R3.ah} = \text{Exp4.a}$
 $\text{Exp3.a} = \text{R3.a}$
 $\text{R3} \rightarrow \text{Op3 Exp4 R3}$
 $\text{R3}_1.\text{ah} = \text{mkexpb}(\text{Op3.op}, \text{R3}_0.\text{ah}, \text{Exp4.a})$
 $\text{R3}_0.\text{a} = \text{R3}_1.\text{a}$
 $\text{R3} \rightarrow \varepsilon$
 $\text{R3.a} = \text{R3.ah}$
 $\text{Exp4} \rightarrow \underline{\text{MINUS}} \text{Exp4}$
 $\text{Exp4}_0.\text{a} = \text{mkexpu}(\text{'-'}, \text{Exp4}_1.\text{a})$
 $\text{Exp4} \rightarrow \underline{\text{NOT}} \text{Exp5}$
 $\text{Exp4.a} = \text{mkexpu}(\text{'not'}, \text{Exp5.a})$
 $\text{Exp4} \rightarrow \text{Exp5}$
 $\text{Exp4.a} = \text{Exp5.a}$
 $\text{Exp5} \rightarrow \underline{\text{LREAL}}$
 $\text{Exp5.a} = \text{num}(\text{LREAL.lex})$
 $\text{Exp5} \rightarrow \underline{\text{TRUE}}$
 $\text{Exp5.a} = \text{true}(\text{TRUE.lex})$
 $\text{Exp5} \rightarrow \underline{\text{FALSE}}$

Exp5.a = false(FALSE.lex)
 Exp5 \rightarrow POP Exp0 PCL
 Exp5.a = Exp0.a
 Op0 \rightarrow PLUS
 Op0.op = '+'
 Op0 \rightarrow MINUS
 Op0.op = '-'
 Op2 \rightarrow EQ
 Op2.op = '=='
 Op2 \rightarrow GT
 Op2.op = '>'
 Op2 \rightarrow GEQ
 Op2.op = '>= '
 Op2 \rightarrow LT
 Op2.op = '< '
 Op2 \rightarrow LEQ
 Op2.op = '<= '
 Op2 \rightarrow NEQ
 Op2.op = '!= '
 Op3 \rightarrow MUL
 Op3.op = '* '
 Op3 \rightarrow DIV
 Op3.op = '/'