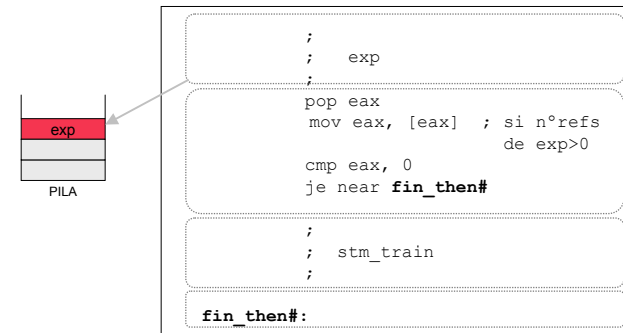


- Generación de código para sentencias condicionales
- Generación de código para bucles while

cond_stm : TOK_IF exp TOK_THEN ★ stm_train TOK_FI ☆



- Las ideas básicas son:
 - modificar ligeramente la gramática.
 - asegurar que el número de etiqueta(#) sea el mismo en las dos veces que aparece **fin_then#** . Para ello se añade un **nuevo atributo semántico**, etiqueta.

cond_stm : TOK_IF exp TOK_THEN ★ stm_train TOK_FI ☆

MODIFICACIÓN DE LA GRAMÁTICA

cond_stm : if_exp_then stm_train TOK_FI ☆
if_exp_then : TOK_IF exp TOK_THEN ★

Comprobaciones semánticas

comprobar que \$2.tipo == TIPO_BOOL
comprobar \$2.nrefs<2

Reserva de etiqueta (#)

\$\$.etiqueta=etiqueta++

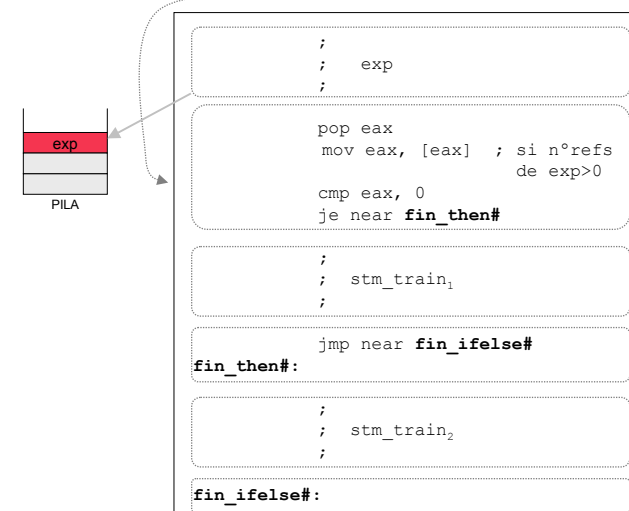
Generación de código

pop eax
mov eax, [eax] ; si n°refs de exp>0
cmp eax, 0
je near **fin_then#** ; # = \$\$.etiqueta

Generación de código

fin_then# : ; # = \$1.etiqueta

cond_stm: TOK_IF exp TOK_THEN ★ stm_train₁ ☆ TOK_ELSE stm_train₂ TOK_FI ☆



Generación de código para sentencias condicionales (IV)

cond_stm: TOK_IF exp TOK_THEN ★ stm_train1 ★ TOK_ELSE stm_train2 TOK_FI ★

MODIFICACIÓN DE LA GRAMÁTICA

```
cond_stm      : if_exp_then stm_train1 TOK_ELSE stm_train2 TOK_FI ★
if_exp_then  : if_exp_then stm_train1 ★
if_exp_then  : TOK_IF exp TOK_THEN ★
```

Comprobaciones semánticas

comprobar que \$2.tipo == TIPO_BOOL
comprobar \$2.nrefs<2

Reserva de etiqueta (#)

\$\$etiqueta=etiqueta++

Generación de código

```
pop eax
mov eax, [eax] ; si n°refs de exp>0
cmp eax, 0
je near fin_then# ; # = $$etiqueta
```

Propagación de atributos

\$\$etiqueta = \$1.etiqueta

Generación de código

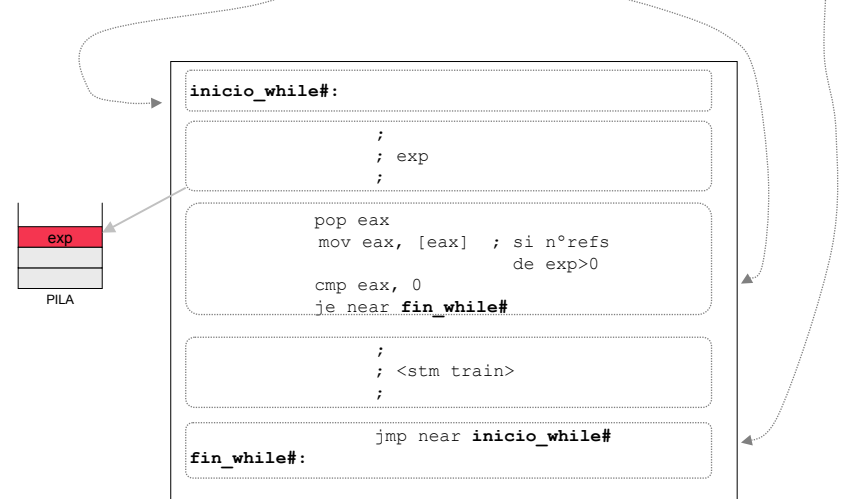
```
jmp near fin_ifelse#
fin_then# :
; # = $1.etiqueta
```

Generación de código

```
fin_ifelse# : ; # = $1.etiqueta
```

Generación de código para bucles while (I)

loop_stm : TOK_WHILE ★ exp TOK_DO ★ stm_train TOK_END ★



Generación de código para bucles while (II)

loop_stm : TOK_WHILE ★ exp TOK_DO ★ stm_train TOK_END ★

MODIFICACIÓN DE LA GRAMÁTICA

```
loop_stm : while_exp_do stm_train TOK_END ★
while_exp_do : token_while exp TOK_DO ★
token_while : TOK_WHILE ★
```

Reserva de etiqueta (#)

\$\$etiqueta=etiqueta++

Generación de código

```
inicio_while# :
; # = $$etiqueta
```

Comprobaciones semánticas

comprobar que \$2.tipo == TIPO_BOOL
comprobar \$2.nrefs<2

Propagación de atributos

\$\$etiqueta = \$1.etiqueta

Generación de código

```
pop eax
mov eax, [eax] ; si n°refs de exp>0
cmp eax, 0
je near fin_while# ; # = $$etiqueta
```

Generación de código

```
jmp near inicio_while#
fin_while# :
; # = $1.etiqueta
```