

Syntax Tree : synthesised attribute n (made in the syntax tree)

program \rightarrow block { return block. n }

block \rightarrow { stmts } { block. n = stmts. n }

stmts \rightarrow stmts₂ stmt { stmts. n = new Seq (stmts₂. n , stmt. n) }

stmts \rightarrow ϵ { stmts. n = null }

stmt \rightarrow expr; { stmt. n = new Eval (expr. n) }

stmt \rightarrow if (expr) stmt₂ { stmt. n = new If (expr. n , stmt₂. n) }

stmt \rightarrow while (expr) stmt₂ ;

stmt \rightarrow do stmt₂ while (expr)

stmt \rightarrow block { stmt. n = block. n }

expr \rightarrow id = expr₂; { expr. n = new Assign (id.entry, expr₂. n) }

expr \rightarrow rel (

rel \rightarrow rel₂ < add /

rel \rightarrow rel <= add \

add \rightarrow add₂ + term (

add \rightarrow term)

term \rightarrow term₂ * factor

factor \rightarrow num { factor. n = new Num (num.value) }

factor \rightarrow (expr) ;

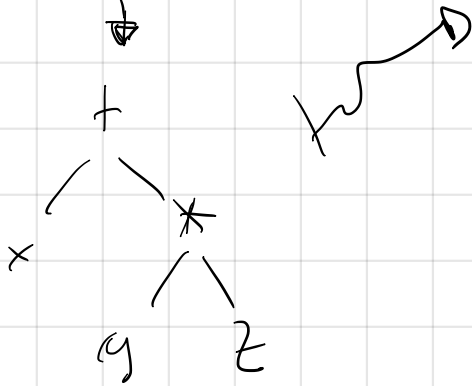
Three-address code

$$x + y * z \rightsquigarrow t_1 = y * z$$

⏟

⌞

$$x = x + t_1$$



param x_1

param x_2

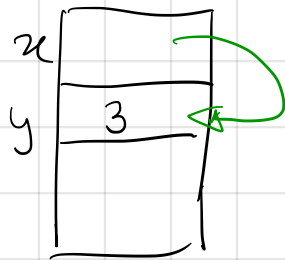
:

param x_m

call p, n or

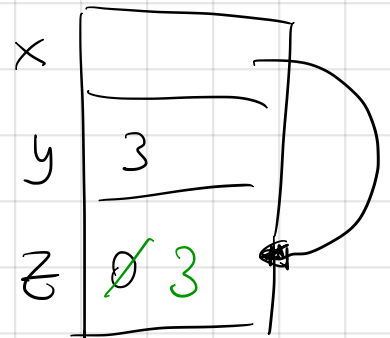
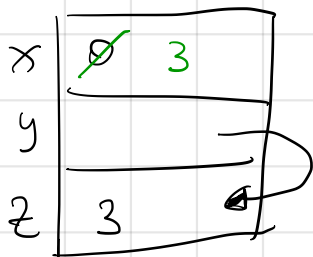
$y = \text{call } p, n$

$$x = \& y$$



$$* x = y$$

$$x = * y$$



do $i = i + 1$; while ($a[i] < v$);

Suppose a : $\text{long}[N]$

long takes 8 bytes

L: $t_1 = i + 1$

$i = t_1$

$t_2 = i * 8$

$t_3 = a[t_2]$

if $t_3 < v$ goto L

:

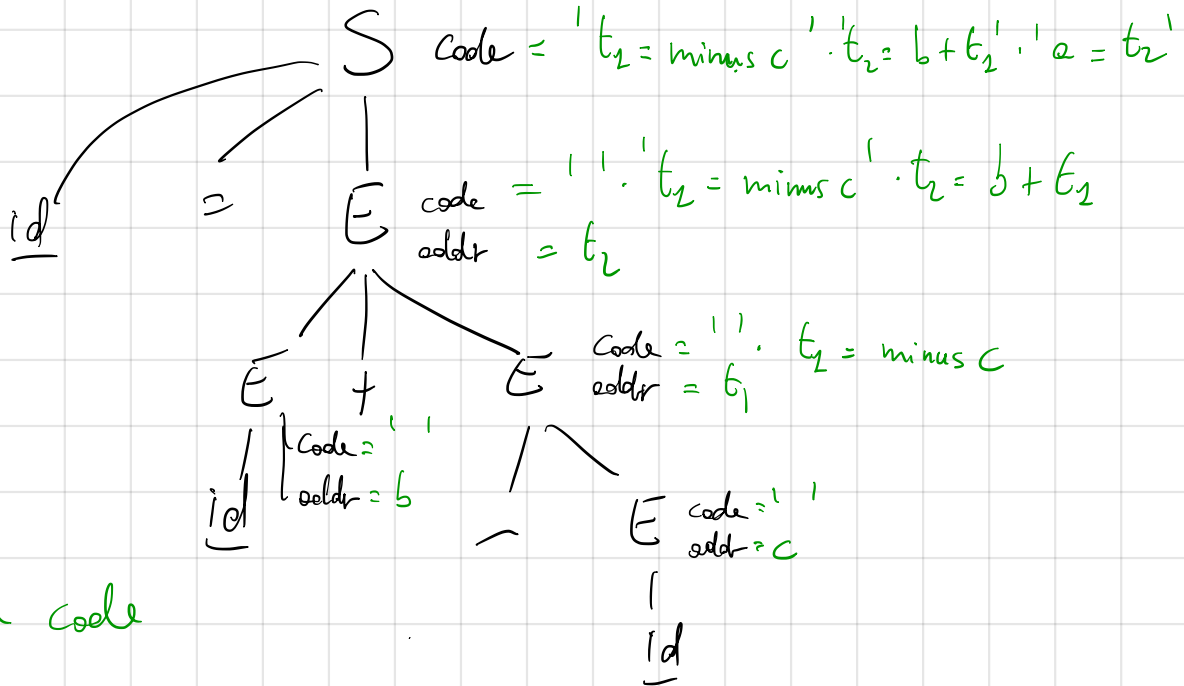
top: ENV

id	type	
x	int	0
y	int	4

int takes 4 bytes

p

$$a = b + -c$$



generated code

$$t_2 = \text{minus } c$$

$$t_2 = b + t_2$$

$$a = t_2$$