#definec *cadena*$_1$ *cadena*$_2$ ➡ ε

Efecto secundario

Hasta *EOF*, cualquier aparición de *cadena* se sustituye '*cadena*$_1$' por '*cadena*$_2$', siempre que '*cadena*$_1$' no forme parte de un '*enombre*' mayor.

Ejemplo

```
#definec foo bar
#definec baz 5

fooz = *foo;
foo$3 = foo + baz;
```

⇒

```
#definec baz 5

fooz = *bar;
foo$3 = bar + baz;
```

⇒

```
fooz = *bar;
foo$3 = bar + 5;
```

#definecc *cadena*$_1$ *cadena*$_2$ ➡ ε

Efecto secundario

Se sustituye igual que en #definec menos en *texto*$_1$ en las siguientes situaciones:

```
){                  :{
   texto1     ó        texto1
}                   }
```

Ejemplo 1

```
#definecc foo bar
:def_subp:{
   foo = 20;
   ...
}
foo = 9;
```

⇒

```
:def_subp:{
   foo = 20;
   ...
}
bar = 9;
```

Ejemplo 1'

```
#definecc bar qux
foo(){
   #definecc bar quux
   bar = 26;
   ...
}
baz(){
   bar = 9;
   ...
}
```

⇒

```
foo(){
   #definecc bar quux
   bar = 26;
   ...
}
baz(){
   bar = 9;
   ...
}
```

⇒

```
foo(){
   quux = 26;
   ...
}
baz(){
   bar = 9;
   ...
}
```

nombre → *identificador de K&R*

⟨*dims*⟩ → ε | **$***n*⟨*dims*⟩

enombre → *nombre*⟨*dims*⟩

⟨*stars*⟩ → ε | ***⟨*stars*⟩

⟨*inds*⟩ → ε | **[***n***]**⟨*inds*⟩

var → ⟨*stars*⟩*nombre*⟨*inds*⟩ | ⟨*stars*⟩*nombre***[]**

⟨*signo*⟩ → + | - | ε

⟨*racional*⟩ → **[***n***$***m***]**

litp → **0** | ⟨*natural*⟩ | ⟨*racional*⟩

lit → ⟨*sign*⟩*litp*

list_lit → ε | *lit* | *lit***,** *list_lit*

⟨*asig*⟩ → *lit* | **{***list_lit***}**

⟨*inic*⟩ → ε | = ⟨*asig*⟩

vari → *var* ⟨*inic*⟩

list_inic → *vari* | *vari***,** *list_inic*

⟨*signed*⟩ → **signed** | ε

type → **unsigned int** | ⟨*signed*⟩ **int** | **float**

identificador de K&R
letra o '_' seguido de cualquier cantidad de símbolos *n*, letras o '_'.

```
signed ➡ ε
```

```
int ➡ float
```

:num_position:

El expansor asocia un ⟨*natural*⟩ a **:num_position:**.

- Antes de empezar las expansiones le asocia 1.
- Si está asociado *m*, después de expandir una intrucción '*unsigned int* ⟨*stars*⟩*nombre*⟨*dims*⟩' asocia *n*, siendo verdad $n = m + 1$.

```
unsigned int ⟨stars_1⟩nombre_1⟨dims_1⟩;
```

⬇

```
#definec nombre_1 Y_n⟨dims_1⟩
#definec nombre_1⟨dims_1⟩ Y_n
```

donde *n* es el ⟨*natural*⟩ asociado a **:num_position:**.

Ejemplo

```
1 INIT
2 unsigned int *foo;
3 unsigned int bar;
4 foo = 47;
5 bar = *foo;
6
```

⇒

```
1 1
2 unsigned int *foo;
3 unsigned int bar;
4 foo = 47;
5 bar = *foo;
6
```

⇒

```
1 1
2 #definec foo Y_1
3 #definec foo Y_1
4 unsigned int bar;
5 foo = 47;
6 bar = *foo;
```

⇒ … ⇒

```
1 1
2 #definec bar Y_2
3 #definec bar Y_2
4 Y_1 = 47;
5 bar = *Y_1;
6
```

⇒ …

```
float ⟨stars1⟩nombre1⟨dims1⟩;
```

⬇

```
unsigned int ⟨stars1⟩nombre1⟨dims1⟩;
#definec ⟨stars1⟩Yn ⟨stars1⟩Fn
```

donde n es el ⟨*natural*⟩ asociado a **:num_position:**.

Ejemplo

```
1 INIT
2 float foo;
3 float **bar[2];
4 bar[1] = 91;
5 *bar[1] = 127;
6 foo = **bar[1];
```

⇒

```
1 1
2 float foo;
3 float **bar[2];
4 bar[1] = 91;
5 *bar[1] = 127;
6 foo = **bar[1];
```

⇒

```
1 1
2 unsigned int foo;
3 #definec Y1 F1
4 float **bar[2];
5 bar[1] = 91;
6 *bar[1] = 127;
7 foo = **bar[1];
```

⇒

```
1 1
2 #definec foo Y1
3 #definec foo Y1
4 #definec Y1 F1
5 float **bar[2];
6 bar[1] = 91;
7 *bar[1] = 127;
8 foo = **bar[1];
```

⇒

```
1 1
2 #definec foo Y1
3 #definec Y1 F1
4 float **bar[2];
5 bar[1] = 91;
6 *bar[1] = 127;
7 Y1 = **bar[1];
```

⇒

```
1 1
2 #definec Y1 F1
3 float **bar[2];
4 bar[1] = 91;
5 *bar[1] = 127;
6 Y1 = **bar[1];
```

⇒

```
1 1
2 float **bar[2];
3 bar[1] = 91;
4 *bar[1] = 127;
5 F1 = **bar[1];
```

⇒ ...

$type_1$ $vari_1$, $list\_inic_1$; ➡

$type_1$ $vari_1$;
$type_1$ $list\_inic_1$;

$type_1$ ⟨$stars_1$⟩$enombre_1$ = lit_1; ➡

$type_1$ ⟨$stars_1$⟩$enombre_1$;
$enombre_1$ = lit_1;

Ejemplo

```
unsigned int foo = 7, bar[3] = {5, 23, 17};
```

⇒

```
unsigned int foo = 7;
unsigned int bar[3] = {5, 23, 17};
```

⇒

```
unsigned int foo;
foo = 7;
unsigned int bar[3] = {5, 23, 17};
```

⇒ ...

Vectores

$type_1$ ⟨$stars_1$⟩$enombre_1$[1]; ➡ $type_1$ ⟨$stars_1$⟩$enombre_1$;

$type_1$ ⟨$stars_1$⟩$enombre_1$[n_1];

⬇

$type_1$ ⟨$stars_1$⟩$enombre_1$[m];
$type_1$ $enombre_1$\$$n_1$;

donde $n = m + 1$ es verdad.

Ejemplo

```
int *foo[3];
```

⇒

```
int *foo[2];
int foo$3;
```

⇒

```
int *foo[1];
int foo$2;
int foo$3;
```

⇒

```
int *foo;
int foo$2;
int foo$3;
```

$type_1\ \langle stars_1 \rangle enombre_1[1] = \{\};$

⬇

$type_1\ \langle stars_1 \rangle enombre_1 = 0;$

$type_1\ \langle stars_1 \rangle enombre_1[n] = \{\};$

⬇

$type_1\ \langle stars_1 \rangle enombre_1 = 0;$
$type_1\ \langle stars_1 \rangle enombre_1\$\$m = \{\};$

donde:

- n no es 1.
- $n = m + 1$ es verdad.

$type_1\ \langle stars_1 \rangle enombre_1[n_1] = \{lit_1\};$

⬇

$type_1\ \langle stars_1 \rangle enombre_1[n_1] = \{lit_1,\};$

$type_1\ \langle stars_1 \rangle enombre_1[n] = \{lit_1,\ list\_lit_1\};$

⬇

$type_1\ \langle stars_1 \rangle enombre_1 = lit_1;$
$type_1\ \langle stars_1 \rangle enombre_1\$\$m = \{list\_lit_1\};$

donde $n = m + 1$ es verdad.

$type_1\ \langle stars_1 \rangle enombre_1\$\$n_1 = \{lit_1\};$

⬇

$type_1\ \langle stars_1 \rangle enombre_1\$\$n_1 = \{lit_1,\};$

$type_1\ \langle stars_1 \rangle enombre_1\$\$1 = \{\};$ ➡ $type_1\ \langle stars_1 \rangle enombre_1\$1 = 0;$

$type_1\ \langle stars_1 \rangle enombre_1\$\$1 = \{lit_1,\};$ ➡ $type_1\ \langle stars_1 \rangle enombre_1\$1 = lit_1;$

$type_1\ \langle stars_1 \rangle enombre_1\$\$n_1 = \{\};$

⬇

$type_1\ \langle stars_1 \rangle enombre_1\$n_1 = 0;$
$type_1\ \langle stars_1 \rangle enombre_1\$\$m = \{\};$

donde:

- n_1 no es 1.
- $n_1 = m + 1$ es verdad.

$type_1\ \langle stars_1 \rangle enombre_1\$\$n_1 = \{lit_1, list\_lit_1\};$

⬇

$type_1\ \langle stars_1 \rangle enombre_1\$\$n_1 = lit_1;$

$type_1\ \langle stars_1 \rangle enombre_1\$\$m = \{list\_lit_1\};$

donde:

- n_1 no es 1.
- $n_1 = m + 1$ es verdad.

Ejemplo 1

```
int foo[3] = {5, 23, 17};
```

⇒

```
int foo = 5;
int foo$$2 = {23, 17};
```

⇒

```
int foo;
foo = 5;
int foo$$2 = {23, 17};
```

⇒ ... ⇒

```
...
int foo$2 = 23;
int foo$$1 = {17};
```

⇒

```
...
int foo$2;
foo$2 = 23;
int foo$$1 = {17};
```

⇒ ... ⇒

```
...
int foo$$1 = {17};
```

⇒

```
...
int foo$$1 = {17,};
```

⇒

```
...
int foo$1 = 17;
```

Ejemplo 1'

```
1 INIT
2 float foo;
3 float **bar[3] = {5, 12};
4 bar[1] = 91;
5 *bar[1] = 127;
6 foo = **bar[1];
```

⇒ ... ⇒

```
1 1
2 float **bar[3] = {5, 12};
3 bar[1] = 91;
4 *bar[1] = 127;
5 F1 = **bar[1];
```

⇒

```
1
float **bar = 5;
float **bar$$2 = {12};
bar[1] = 91;
*bar[1] = 127;
F1 = **bar[1];
```

⇒

```
1
float **bar;
bar = 5;
float **bar$$2 = {12};
bar[1] = 91;
*bar[1] = 127;
F1 = **bar[1];
```

⇒

```
1
unsigned int **bar;
#definec **Y2 **F2
bar = 5;
float **bar$$2 = {12};
bar[1] = 91;
*bar[1] = 127;
F1 = **bar[1];
```

⇒

```
1
#definec bar Y2
#definec bar Y2
#definec **Y2 **F2
bar = 5;
float **bar$$2 = {12};
bar[1] = 91;
*bar[1] = 127;
F1 = **bar[1];
```

⇒

```
1
#definec bar Y2
#definec **Y2 **F2
Y2 = 5;
float **bar$$2 = {12};
Y2[1] = 91;
*Y2[1] = 127;
F1 = **Y2[1];
```

⇒

```
1
#definec **Y2 **F2
Y2 = 5;
float **bar$$2 = {12};
Y2[1] = 91;
*Y2[1] = 127;
F1 = **Y2[1];

```

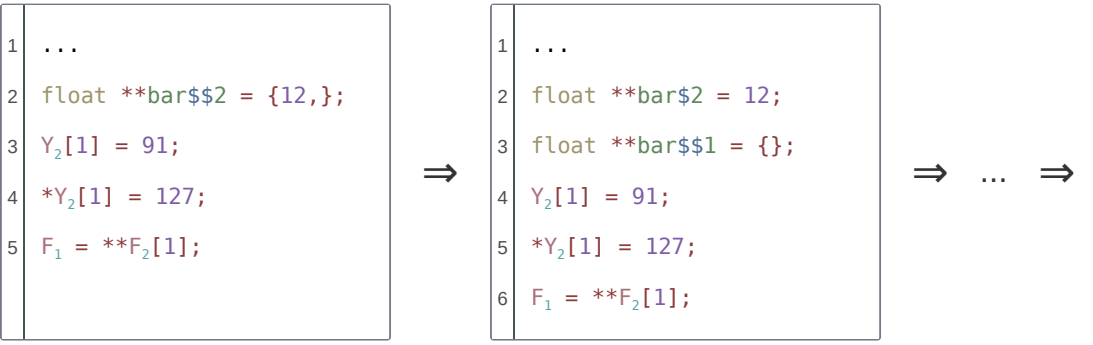
⇒

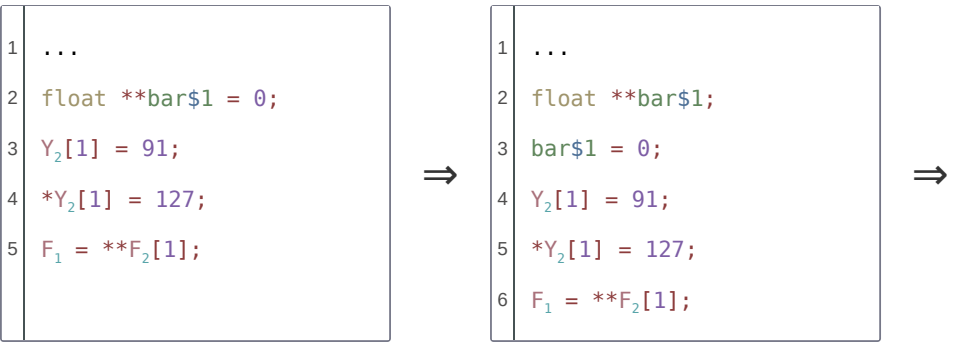
```
1
Y2 = 5;
float **bar$$2 = {12};
Y2[1] = 91;
*Y2[1] = 127;
F1 = **F2[1];
```

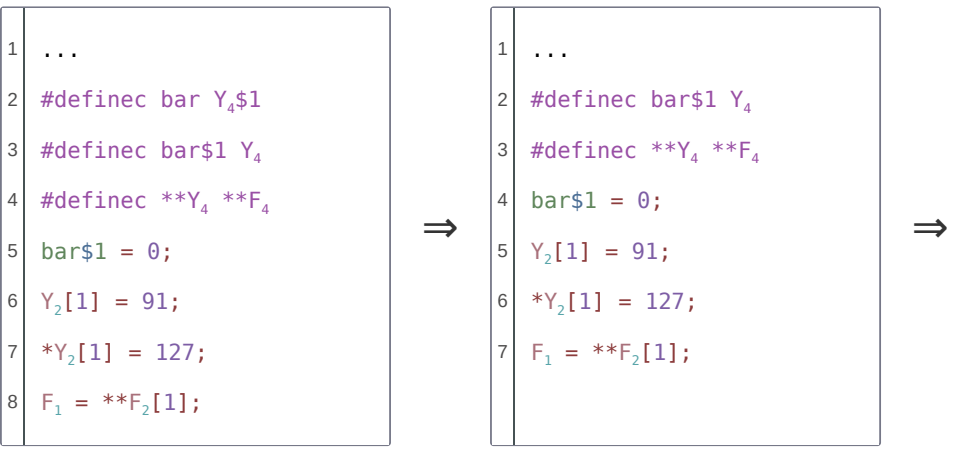
⇒ ... ⇒

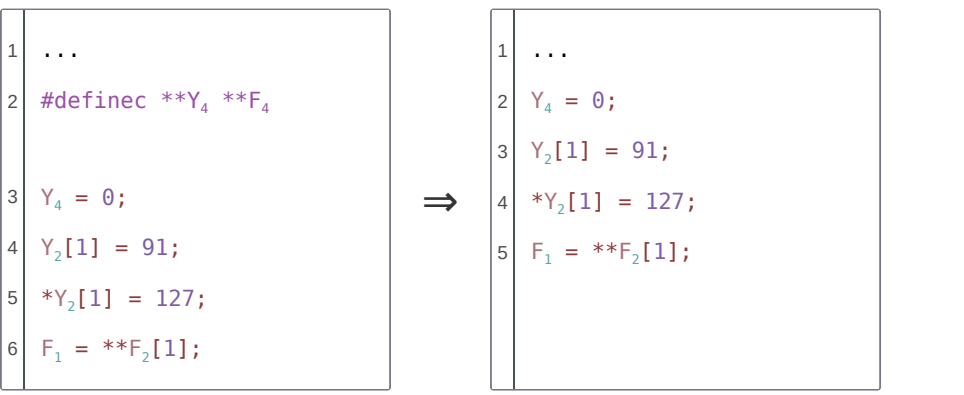
```
...
float **bar$$2 = {12};
Y2[1] = 91;
*Y2[1] = 127;
F1 = **F2[1];
```

⇒


...
float **bar$$2 = {12,};
Y2[1] = 91;
*Y2[1] = 127;
F1 = **F2[1];
⇒
...
float **bar$2 = 12;
float **bar$$1 = {};
Y2[1] = 91;
*Y2[1] = 127;
F1 = **F2[1];
⇒ ... ⇒


...
float **bar$1 = 0;
Y2[1] = 91;
*Y2[1] = 127;
F1 = **F2[1];
⇒
...
float **bar$1;
bar$1 = 0;
Y2[1] = 91;
*Y2[1] = 127;
F1 = **F2[1];
⇒


...
#definec bar Y4$1
#definec bar$1 Y4
#definec **Y4 **F4
bar$1 = 0;
Y2[1] = 91;
*Y2[1] = 127;
F1 = **F2[1];
⇒
...
#definec bar$1 Y4
#definec **Y4 **F4
bar$1 = 0;
Y2[1] = 91;
*Y2[1] = 127;
F1 = **F2[1];
⇒


...
#definec **Y4 **F4
Y4 = 0;
Y2[1] = 91;
*Y2[1] = 127;
F1 = **F2[1];
⇒
...
Y4 = 0;
Y2[1] = 91;
*Y2[1] = 127;
F1 = **F2[1];

$type_1\ \langle stars_1 \rangle enombre_1[n][m_1]$

⬇

$type_1\ \langle stars_1 \rangle enombre_1\$m_1[p]$

donde $p = m * n$ es verdad.

Ejemplo

```
int foo[4][2][2] = {81, 11,  8,  3,
                    19, 53,  1, 32,
                    12, 82, 13, 13,
                     9,  2,  1, 54};
```

⇒

```
int foo$2[8][2] = {81, 11,  8,  3,
                   19, 53,  1, 32,
                   12, 82, 13, 13,
                    9,  2,  1, 54};
```

⇒

```
int foo$2$2[16] = {81, 11,  8,  3,
                   19, 53,  1, 32,
                   12, 82, 13, 13,
                    9,  2,  1, 54};
```

⇒

```
int foo$2$2[16] = {81, 11,  8,  3,
                   19, 53,  1, 32,
                   12, 82, 13, 13,
                    9,  2,  1, 54};
```

⇒

```
int foo$2$2 = 81;
int foo$2$2$$15 = {11,  8,  3, 19,
                   53,  1, 32, 12,
                   82, 13, 13,  9,
                    2,  1, 54};
```

$type_1\ \langle stars_1 \rangle nombre_1[] = \{lit_1\};$ ➡ $type_1\ \langle stars_1 \rangle enombre_1[] = \{lit_1,\};$

$type_1\ \langle stars_1 \rangle nombre_1[] = \{lit_1,\ list\_lit_1\};$

⬇

$type_1\ \langle stars_1 \rangle nombre_1 = lit_1;$
$type_1\ \langle stars_1 \rangle nombre_1\$1 = \{list\_lit_1\};$

$type_1\ \langle stars_1 \rangle nombre_1\$n = \{\};$ ➡ ε

$type_1\ \langle stars_1 \rangle nombre_1\$n = \{lit_1\};$ ➡ $type_1\ \langle stars_1 \rangle nombre_1\$n = \{lit_1,\};$

$type_1\ \langle stars_1 \rangle nombre_1\$n = \{lit_1,\ list\_lit_1\};$

⬇

$type_1\ \langle stars_1 \rangle nombre_1\$n = lit_1;$
$type_1\ \langle stars_1 \rangle nombre_1\$m = \{list\_lit_1\};$

donde $m = n + 1$ es verdad.

Ejemplo

```
...
int foo[] = {55, 24, 8,}
foo[2] = 81;
```

⇒

```
...
int foo = 55;
int foo$1 = {24, 8,}
foo[2] = 81;
```

⇒ ... ⇒

```
...
int foo$1 = 24;
int foo$2 = {8,}
F9[2] = 81;
```

⇒ ... ⇒

```
...
int foo$2 = 8;
int foo$3 = {};
F9[2] = 81;
```

⇒ ... ⇒

```
...
F11 = 8;
int foo$3 = {};
F9[2] = 81;
```

⇒

```
...
1^423
int foo$3 = {};
F9[2] = 81;
```

⇒

```
...
1^423
F9[2] = 81;
```

$type_1\ \langle stars_1 \rangle enombre_1[][m_1]$ ➡ $type_1\ \langle stars_1 \rangle enombre_1\$m_1[]$

Ejemplo

```
int foo[][2][2] = {81, 11,  8,  3,
                   19, 53,  1, 32,
                   12, 82, 13, 13}
```

⇒

```
int foo$2[][2] = {81, 11,  8,  3,
                  19, 53,  1, 32,
                  12, 82, 13, 13}
```

⇒

```
int foo$2$2[] = {81, 11,  8,  3,
                 19, 53,  1, 32,
                 12, 82, 13, 13}
```

Asignaciones

$\langle opd \rangle \rightarrow$ ++ | --

$\langle yd \rangle \rightarrow \langle opd \rangle \langle stars \rangle \mathbf{Y}_n \mid \langle stars \rangle \mathbf{Y}_n \langle opd \rangle$

$\langle vindx \rangle \rightarrow \langle natural \rangle \mid \mathbf{Z}_\Omega \mid \langle stars \rangle \mathbf{Y}_n \mid \langle yd \rangle \mid \mathbf{0}$

$\langle indx \rangle \rightarrow \mathbf{[} \langle vindx \rangle \mathbf{]}$

$\langle indxs \rangle \rightarrow \varepsilon \mid \langle indx \rangle \langle indxs \rangle$

$V \rightarrow \mathbf{Y} \mid \mathbf{F}$

$V'_n \rightarrow \langle stars \rangle V_n \langle dims \rangle \langle indxs \rangle$

$\langle preinstrucción \rangle \rightarrow x^\Omega$

$\langle preinstrucciones \rangle \rightarrow \varepsilon \mid x^\Omega \langle preinstrucciones \rangle$

Expansiones finales 1'

Cuando se usan macros de la forma '*unsigned int* ⟨*stars*⟩*nombre*⟨*dims*⟩' el expansor, en las expansiones finales:

- después de agregar las instrucciones '1', añade al final una nueva instrucción $1^{:top}$ con una marca $\leftarrow$**:*i*** apuntando a ella.
- sustituye todas las apariciones de **:top** por **:*i*** en todas las $x^{:top}$.