```
let g = proc (int x, int y, bool z, bool w)
            if <(x, y) then and(z, w)
            else w
    { f = proc (int * int * bool * bool → bool h,
               int → bool p, int m, int n, bool o)
            if (p m) then false
            else (h n m (p n) (p * (2, m)))
    { q = proc (int a)
            > (a, 4)
      i = false
in
  (f g q 1 5 i)
```

(p m)

$t_p = t_m \to bool$

$t_p = int \to bool$

$t_m = int$

(h n m (p n) (p * (2, m))

(p n) $t_n = int$

(p * (2, m))

int

(f g q 1 5 i)

$t_f = t_g * t_q * int * int * bool$

$t_f = (int * int * bool * bool \to bool) * (int \to bool) * int * int * bool \to bool$

$t_g = (int * int * bool * bool \to bool)$

$t_g = (int * int * bool * bool \to bool)$

<(x, y)

$t_x * t_y \to bool$ $t_x = int$

$int * int \to bool$ $t_y = int$

and(z, w)

$t_z * t_w \to bool$

$bool * bool \to bool$

$t_q = int \to bool$

$t_q \to int \to bool$ $t_q = int$

(b) **[26 pts.]** El tipo de la expresión

```
let g = proc (x, y, z, t, w)
            if (z y t) then (x true t)
            else (x false w)
    f = proc (j, k)
            if j then *(k, k) else k
    a = 2
in
  let fun1 = proc (l, m)
                if >(l, m) then true
                else false
  in
    let v2 = (g f 3 fun1 4 *(2, a))
    in
      (fun1 v2 (f true a))
```

Exp	Vtp
g	tg
x	tx
y	ty
z	tz
t	tt
w	tw
f	tf
j	tj
k	tk
a	ta
fun1	tfun1
l	tl
m	tm
v2	tv2

tp bool

(z y t) bool t1

(x true t) int t2

(x false w) int t3

*(k, k) int t4

>(l, m) bool t5

*(z, a) int t6

(g f 3 fun1 4 *(z, a)) t7 int

(f true a) t8 int

tg = (bool * int → int) * int * (int * int → bool) * int * int → int

$v_2 = (g\ f\ 3\ fun_1\ 4\ *\ (2, a))$

$t_g = t_f * int * t_{fun1} * int * t_6 \to t_7$

$v_2 = t_7$

$g = \textbf{proc}\ (x, y, z, t, w)$ t_2
$\quad \textbf{if}\ (z\ y\ t)\ \textbf{then}\ (x\ true\ t)$
$\quad \textbf{else}\ (x\ false\ w)$: t_3

$t_g = t_x * t_y * t_z * t_t * t_w \to t_2$

$t_g = t_x * t_y * t_z * t_t * t_w \to t_3$

$t_g = (bool * int \to int) * int * (int * int \to bool) * int * int \to int$

$(z\ y\ t)$

$t_z = t_y * t_t \to bool$

- $t_z = int * int \to bool$

$(x\ true\ t)$

$t_x = bool * t_t \to t_2$

$t_x = bool * int \to t_2$

$(x\ false\ w)$: t_3

$t_x = bool * t_w \to t_3$

$a = 3$

$t_a = int$

- $t_x = t_f$
  - $t_y = int$
  - $t_z = t_{fun1}$
    - $t_t = int$
  - $int = t_w = t_6$
  - $t_2 = t_3$
  - $t_7 = t_2 = t_3$
  - $t_x = bool * int \to int$
  - $t_2 = int$
  - $t_3 = int$
  - $t_7 = int$
  - $t_w = int$
  - $t_{v2} = int$

$(f\ true\ a)$

$t_F = bool \times t_a \to t_8$

$t_F = bool \times int \to t_8$

↘ Google colab

$f = \mathbf{proc}\ (j, k)$
$\quad \mathbf{if}\ j\ \mathbf{then}\ *(k, k)\ \mathbf{else}\ k$

$t_F = t_j \times t_K \to t_K$ (K 2)

$t_F = t_j \times t_K \to t_4$

$*(K, K)$

$t_{K*} \times t_K \to t_4$

$int \times int \to int$

$t_j = bool$

$t_K = int$

$t_K = t_4 = int$

$t_F = bool \times int \to int$

$t_4 = t_8$

$t_8 = int$

$(fun_1\ v_2\ (f\ true\ a))$

$t_{Fun_1} = t_{V2} \times t_8 \to t_p$

$t_{Fun1} = t_{V2} \times int \to t_p$

$t_{fun1} = int \times int \to t_p$

$fun_1 = \mathbf{proc}\ (l, m)$
$\quad \mathbf{if}\ >(l, m)\ \mathbf{then}\ true$
$\quad \mathbf{else}\ false$

$t_{Fun_1} = t_l \times t_m \to bool$

$t_{Fun1} = int \times int \to t_p$

$>(l, m)$

$t_l \times t_m \to bool$

$int \times int \to bool$

$t_l \times t_m \to t_5$

$t_p = bool$

$t_l = int$

$t_m = int$