```
let
  f = proc (?x, int y)
      if x then +(y, 1)
      else -(y,1)
in
  let
    g = proc (?m, int n)
        (m true n)
  in
    let
      h = (g f 5)
    in
      g
```

$t_f = bool * int \to int$

$t_x = bool$

$t_y = int$ ✓

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

$t_m = bool * int \to int$

$t_n = int$

$t_h = int$

$t_1 = t_g$

tg = ((bool * int -> int) * int -> int)

1) Proc

$t_f = t_x * t_y \to t_2$

$t_g = t_m * t_n \to t_3$

$t_h = t_3$

2) app-exp (evaluación)

(m true n)

$t_m = bool * t_n \to t_4$

(g f 5)

$t_g = t_f * int \to t_5$

3) Rel 1 y 2

$t_g = ((bool * t_n) \to t_4) * t_n \to t_3$

$t_f = t_m$

$t_n = int$

$t_5 = t_3 = t_h$

$t_g = ((bool * int) \to t_4) * int \to t_h$

(t_f)

$t_x = bool \quad t_2 = t_4$

$t_y = int$

---

4) Intima proc

```
if x then +(y, 1)   (int)
     else -(y,1)    (int)
```

$t_x = int$

$t_f = bool * int \to int$

(m true n) = t_4

$t_4 = t_3 = t_2 = int$

\- $t_y * int \to int = int * int \to int$

\+ $t_y * int \to int = int * int \to int$

$t_2 = int$

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