```
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_1 t_2 t_g

$t_f, t_x, t_y, t_g, t_m, t_n, t_h$

$$\begin{cases} t_f = t_x * t_y \to t_1 \\ t_g = t_m * t_n \to t_2 \end{cases}$$

Procedimento

dichiarazione proc

$$\begin{cases} t_f = t_x * int \to t_1 \\ t_g = t_m * int \to t_2 \end{cases}$$

- $t_y = int$
- $t_n = int$
- $t_x = bool$
- $t_f = bool * int \to int$
- $t_m = bool * int \to int$

if x then +(y,1) else -(y,1) t_1

$t_x = bool$

$+ \to int * int \to int$

$t_y * int \to int$

$t_y * int \to int$

regola if

$t_f = bool * int \to int$

$t_m = bool * t_n \to t_2$

$t_m = bool * int \to t_2$

$t_2 = int$

$t_h = int$

$t_h = t_2$

$t_g = t_f * int \to t_2$

$t_m = t_f$

$t_2 = int$

$T = t_g$

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

```
let
    f = proc(? x, ? y, ? z)
①     if (x y) then *(z,2) else z
in                        ②
    let
      g = proc(? m)
          if m then true else false
      k = 5 ③
    in
      (f g true k) ④
```

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

$t_x = bool \to bool$

$t_y = bool$

$t_z = int$

$t_g = bool \to bool$

$t_m = bool$

$t_k = int$

$t_F = t_x * t_y * t_z \to t_1$

$t_g = t_m \to t_2$

} declarations

① $t_x = t_y \to bool$

$int * int \to int$

② $t_z * int \to int$

$int = t_z$

③ $t_m = bool$

$t_2 = bool$

④ $t_F = t_g * bool * t_k \to t_1$

$t_g = t_x$

$t_y = bool$

$t_k = t_z$

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