Ejercicio 1

Para cada uno de los siguientes procesos, indicar si están bien tipados o no. En caso afirmativo, mostrar una derivación del juicio de tipado.

a)

```
x+:end, z:int completed                                  x−:end completed
----------------------- [T−Nil]                          ------------------ [T−Nil]    --------- [T−Aux]
x+:end, z:int ⊢ 0                                        x−:end ⊢ 0                    ⌀ ⊢ 2:int
---------------------------------- [T−In]                ---------------------------------------- [T−Out]    --------- [T−Aux]
x+:?int.end, z:int ⊢ x+?(z:int).0                        x−:!int.end ⊢ x−!2.0                                ⌀ ⊢ 1:int
---------------------------------------- [T−In]          ------------------------------------------------------------ [T−Out]
x+:?int.?int.end ⊢ x+?(z:int).x+?(z:int).0               x−:!int.!int.end ⊢ x−!1.x−!2.0
------------------------------------------------------------------------------------------ [T−Par]
x+:?int.?int.end, x−:!int.!int.end ⊢ x+?(z:int).x+?(z:int).0 | x−!1.x−!2.0
------------------------------------------------------------------------------ [T−Res]
⌀ ⊢ (νx:?int.?int.end)(x+?(z:int).x+?(z:int).0 | x−!1.x−!2.0)
```

Bien tipado.

b)

```
                X                                                    X
x+:!int.!int.end ⊢ x+?(z:int).x+?(z:int).0               x−:?int.?int.end ⊢ x−!1.x−!2.0
------------------------------------------------------------------------------------------ [T−Par]
x+:!int.!int.end, x−:?int.?int.end ⊢ x+?(z:int).x+?(z:int).0 | x−!1.x−!2.0
------------------------------------------------------------------------------ [T−Res]
⌀ ⊢ (νx:!int.!int.end)(x+?(z:int).x+?(z:int).0 | x−!1.x−!2.0)
```

No está bien tipado porque por convención el session type declarado en la restricción `νx:S` le corresponde al endpoint `x+`, y luego el session type dual de `S` le corresponde al otro endpoint `x−`. Por lo tanto el juicio de tipado se traba porque los procesos están usando los endpoints al revés.

c)

```
                                                                X
                                                x−:!int.!int.end ⊢ x−!1.0    ⌀ ⊢ x−!2.0
                                                ------------------------------------ [T−Par]
x+:?int.?int.end ⊢ x+?(z:int).x+?(z:int).0    x−:!int.!int.end ⊢ x−!1.0 | x−!2.0
------------------------------------------------------------------------------- [T−Par]
x+:?int.?int.end, x−:!int.!int.end ⊢ x+?(z:int).x+?(z:int).0 | x−!1.0 | x−!2.0
------------------------------------------------------------------------------- [T−Res]
⌀ ⊢ (νx:?int.?int.end)(x+?(z:int).x+?(z:int).0 | x−!1.0 | x−!2.0)
```

No está bien tipado porque 2 procesos quieren usar el mismo endpoint `x−` y ya vimos que eso no es posible con **binary** session types. El proceso `x−!2.0` pierde el tipado del endpoint `x−` y se traba el juicio de tipado.

d)

```
x+:end, z:int completed                              x−:end completed
----------------------- [T−Nil]                      ---------------- [T−Nil]    --------- [T−Aux]
x+:end, z:int ⊢ 0                                    x−:end ⊢ 0                  ⌀ ⊢ 2:int
--------------------------------- [T−In]             ------------------------------------ [T−Out]    --------- [T−Aux]
x+:?int.end, z:int ⊢ x+?(z:int).0                    x−:!int.end ⊢ x−!2.0                            ⌀ ⊢ 1:int
------------------------------------------ [T−In]    ------------------------------------------------------------ [T−Out]
x+:?int.?int.end ⊢ x+?(z:int).x+?(z:int).0           x−:!int.!int.end ⊢ x−!1.x−!2.0
------------------------------------------------------------------------------- [T−Par]
x+:?int.?int.end, x−:!int.!int.end ⊢ x+?(z:int).x+?(z:int).0 | x−!1.x−!2.0
```

Bien tipado. Necesitamos agregar al contexto inicial el tipado del canal `x` .

e)

```
                                                            X
                                                x−:!int.end ⊢ x−!true.0     ⌀ ⊢ 1:int
                                                ------------------------------------ [T−Out]
x+:?int.?int.end ⊢ x+?(z:int).x+?(z:int).0           x−:!int.!int.end ⊢ x−!1.x−!true.0
------------------------------------------------------------------------------- [T−Par]
x+:?int.?int.end, x−:!int.!int.end ⊢ x+?(z:int).x+?(z:int).0 | x−!1.x−!true.0
```

No está bien tipado porque el proceso `x−!true.0` quiere mandar un `bool` pero el endpoint `x−:!int.end` solo admite enviar un `int` .

f)

Sea `P ≡ (νx:?int.?int.end)(x+?(z:int).x+?(z:int).0 | x−!1.z+!x−.0)` .

```
                                          ---------- [T-Aux]
z+:!(!int.end).end, z-:end, w:end ⊢ P     ⌀ ⊢ 2:int
------------------------------------------------- [T-Out]
z+:!(!int.end).end, z-:end, w:!int.end ⊢ P | w!2.0
------------------------------------------------------------- [T-In]
z+:!(!int.end).end, z-:?(!int.end).end ⊢ P | z-?(w:!int.end).w!2.0
```

Basta ver el juicio de tipado de `z+:!(!int.end).end, z-:end, w:end ⊢ P`.

```
z-:end, w:end, x+:end, y:int completed                          z+:end completed
------------------------------------- [T-Nil]                   ---------------- [T-Nil]    ------------------------- [T-Aux]
z-:end, w:end, x+:end, y:int ⊢ 0                                z+:end ⊢ 0                  x-:!int.end ⊢ x-:!int.end
------------------------------------------------ [T-In]         ------------------------------------------------------ [T-Out]    --------- [T-Aux]
z-:end, w:end, x+:?int.end, y:int ⊢ x+?(y:int).0                z+:!(!int.end).end, x-:!int.end ⊢ z+!x-.0                    ⌀ ⊢ 1:int
------------------------------------------------------- [T-In]  ------------------------------------------------------------------------------ [T-Out]
z-:end, w:end, x+:?int.?int.end ⊢ x+?(y:int).x+?(y:int).0       z+:!(!int.end).end, x-:!int.!int.end ⊢ x-!1.z+!x-.0
------------------------------------------------------------------------------------------------------------------------------------------ [T-Par]
z+:!(!int.end).end, z-:end, w:end, x+:?int.?int.end, x-:!int.!int.end ⊢ x+?(z:int).x+?(z:int).0 | x-!1.z+!x-.0
------------------------------------------------------------------------------------------------------------------------------------------- [T-Res]
z+:!(!int.end).end, z-:end, w:end ⊢ (νx:?int.?int.end)(x+?(z:int).x+?(z:int).0 | x-!1.z+!x-.0)
```

Bien tipado.

g)

```
(νx:?int.?int.end)(x+?(z:int).x+?(z:int).0 | x-!1.z+!x-.0) | z-?(w:!int.end).w-!2.0
```

Es muy parecido al inciso f) pero no está bien tipado. Puntualmente en este proceso:

```
z-?(w:!int.end).w-!2.0
```

Recibimos un canal delegado el cual ligamos a la variable `w`. Pero luego se intenta enviar por el canal `w-` con polaridad y esta variable no está en el contexto. Cuando se recibe un canal delegado no le ponemos polaridad a la variable ya que no podemos saber qué endpoint fue delegado.

h)

```
y−:end, x+:end, z:int completed
------------------------------- [T−Nil]
y−:end, x+:end, z:int ⊢ 0
---------------------------------- [T−In]    ------------------------- [T−Aux]
y−:end, x+:?int.end ⊢ x+?(z:int).0          x−:!int.end ⊢ x−:!int.end
------------------------------------------------------------------------ [T−Out]                      X
y−:!(!int.end).end, x+:?int.end, x−:!int.end ⊢ y−!x−.x+?(z:int).0               y+:?(!int.end).end ⊢ y+?(z:?int.end).z?(w:int).0
------------------------------------------------------------------------------------------------------------------------------------ [T−Par]
y−:!(!int.end).end, y+:?(!int.end).end, x+:?int.end, x−:!int.end ⊢ y−!x−.x+?(z:int).0 | y+?(z:?int.end).z?(w:int).0
------------------------------------------------------------------------------------------------------------------------------------ [T−Res]
y−:!(!int.end).end, y+:?(!int.end).end ⊢ (νx:?int.end)(y−!x−.x+?(z:int).0 | y+?(z:?int.end).z?(w:int).0)
```

No está bien tipado. En el contexto tenemos que `y+:?(!int.end).end` recibe un canal delegado con session type `!int.end`. Pero el proceso liga la variable `z` con session type `?int.end`.

i)

```
y−:end, x+:end, z:int completed                                                 y+:end, z:end completed
------------------------------- [T−Nil]                                         ----------------------- [T−Nil]    ---------- [T−Aux]
y−:end, x+:end, z:int ⊢ 0                                                       y+:end, z:end ⊢ 0                   ⌀ ⊢ 1:int
---------------------------------- [T−In]    ------------------------- [T−Aux]  ------------------------------------------ [T−Out]
y−:end, x+:?int.end ⊢ x+?(z:int).0          x−:!int.end ⊢ x−:!int.end           y+:end, z:!int.end ⊢ z!1.0
------------------------------------------------------------------------ [T−Out]  ---------------------------------------- [T−In]
y−:!(!int.end).end, x+:?int.end, x−:!int.end ⊢ y−!x−.x+?(z:int).0               y+:?(!int.end).end ⊢ y+?(z:!int.end).z!1.0
------------------------------------------------------------------------------------------------------------------------------------ [T−Par]
y−:!(!int.end).end, y+:?(!int.end).end, x+:?int.end, x−:!int.end ⊢ y−!x−.x+?(z:int).0 | y+?(z:!int.end).z!1.0
------------------------------------------------------------------------------------------------------------------------------------ [T−Res]
y−:!(!int.end).end, y+:?(!int.end).end ⊢ (νx:?int.end)(y−!x−.x+?(z:int).0 | y+?(z:!int.end).z!1.0)
```

Bien tipado.

Ejercicio 2

Para cada uno de los siguientes procesos, dar una definición de `P` tal que el proceso sea bien tipado.

a)

```
x+?(y:&[Done:end, Next:?int.!int.end]).P
```

```
P ≡ y▷[Done:0, Next:y?(z:int).y!1.0]
```

```
                                        x+:end, y:end, z:int completed
                                 --------- [T-Aux]    ------------------------------ [T-Aux]
                                 ⌀ ⊢ 1:int            x+:end, y:end, z:int ⊢ 0
                                 ------------------------------------------- [T-Out]
x+:end, y:end completed          x+:end, y:!int.end, z:int ⊢ y!1.0
----------------------- [T-Nil]  ---------------------------------------- [T-In]
x+:end, y:end ⊢ 0                x+:end, y:?int.!int.end ⊢ y?(z:int).y!1.0
------------------------------------------------------------------------ [T-Branch]
x+:end, y:&[Done:end, Next:?int.!int.end] ⊢ y▷[Done:0, Next:y?(z:int).y!1.0]
------------------------------------------------------------------------------------------------------------------------ [T-In]
x+:?(&[Done:end, Next:?int.!int.end]).end ⊢ x+?(y:&[Done:end, Next:?int.!int.end]).y▷[Done:0, Next:y?(z:int).y!1.0]
```

b)

```
(νx:⊕[Pr:!int.?bool.end, Co:!int.!int.?bool.end]) (P | Q)
```

```
P = (x+◁Pr.x+!1.x+?(z:bool).0)
Q = (x−▷[Pr:x−?(z:int).x−!true.0, Co:x−?(z:int).x−?(z:int).x−!true.0])
```

```
S = ⊕[Pr:!int.?bool.end, Co:!int.!int.?bool.end]
T = &[Pr:?int.!bool.end, Co:?int.?int.!bool.end]
```

```
(1)         (2)
--------    ---------
x+:S ⊢ P    x−:T ⊢ Q
---------------------- [T-Par]
x+:S, x−:T ⊢ P | Q
------------------- [T-Res]
(νx:S) (P | Q)
```

Rama `(1)`

```
                    x+:end, z:bool completed
                    ------------------------ [T-Nil]
                    x+:end, z:bool ⊢ 0
--------- [T-Aux]   ---------------------------- [T-In]
⌀ ⊢ 1:int           x+:?bool.end ⊢ x+?(z:bool).0
------------------------------------------------- [T-Out]
x+:!int.?bool.end ⊢ x+!1.x+?(z:bool).0
------------------------------------------------------------------------- [T-Choice]
x+:⊕[Pr:!int.?bool.end, Co:!int.!int.?bool.end] ⊢ x+◁Pr.x+!1.x+?(z:bool).0
```

Rama (2)

```
                                                                    x−:end, z:int completed
                                                  ------------ [T-Aux]   ----------------------- [T-Nil]
                      x−:end, z:int completed     ⌀ ⊢ true:bool          x−:end, z:int ⊢ 0
------------ [T-Aux]  ----------------------- [T-Nil]  ---------------------------------------- [T-Out]
⌀ ⊢ true:bool         x−:end, z:int ⊢ 0               x−:!bool.end, z:int ⊢ x−!true.0
---------------------------------------- [T-Out]      ---------------------------------------- [T-In]
x−:!bool.end, z:int ⊢ x−!true.0                       x−:?int.!bool.end, z:int ⊢ x−?(z:int).x−!true.0
---------------------------------------- [T-In]       ------------------------------------------------------ [T-In]
x−:?int.!bool.end ⊢ x−?(z:int).x−!true.0              x−:?int.?int.!bool.end ⊢ x−?(z:int).x−?(z:int).x−!true.0
------------------------------------------------------------------------------------------------------------------------ [T-Branch]
x−:&[Pr:?int.!bool.end, Co:?int.?int.!bool.end] ⊢ (x−▷[Pr:x−?(z:int).x−!true.0, Co:x−?(z:int).x−?(z:int).x−!true.0])
```