

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
----- [T-Nil]
x+:end, z:int ⊢ 0

----- [T-In]
x+:?int.end, z:int ⊢ x+(z:int).0

----- [T-In]
x+:?int.?int.end ⊢ x+(z:int).x+(z:int).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]
∅ ⊢ (vx:?int.?int.end)(x+(z:int).x+(z:int).0 | x-!1.x-!2.0)

x-:end completed
----- [T-Nil]
x-:end ⊢ 0

----- [T-Out]
x-:!int.end ⊢ x-!2.0

----- [T-Out]
x-:!int.!int.end ⊢ x-!1.x-!2.0

----- [T-Aux]
∅ ⊢ 2:int

----- [T-Aux]
∅ ⊢ 1:int
```

Bien tipado.

b)

```

X
x+:!int.!int.end ⊢ x+(z:int).x+(z:int).0

X
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]
∅ ⊢ (vx:!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 `vx: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)

$$\begin{array}{c}
\text{X} \\
x-:!int.!int.end \vdash x-!1.0 \quad \emptyset \vdash x-!2.0 \\
\hline
\text{[T-Par]} \\
x+:?int.?int.end \vdash x+(z:int).x+(z:int).0 \quad x-:!int.!int.end \vdash x-!1.0 \mid x-!2.0 \\
\hline
\text{[T-Par]} \\
x+:?int.?int.end, x-:!int.!int.end \vdash x+(z:int).x+(z:int).0 \mid x-!1.0 \mid x-!2.0 \\
\hline
\text{[T-Res]} \\
\emptyset \vdash (vx:?int.?int.end)(x+(z:int).x+(z:int).0 \mid x-!1.0 \mid x-!2.0)
\end{array}$$

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)

$ \begin{array}{c} x+:end, z:int \text{ completed} \\ \hline \text{[T-Nil]} \\ x+:end, z:int \vdash \emptyset \\ \hline \text{[T-In]} \\ x+:?int.end, z:int \vdash x+(z:int).0 \\ \hline \text{[T-In]} \\ x+:?int.?int.end \vdash x+(z:int).x+(z:int).0 \\ \hline \text{[T-Par]} \\ x+:?int.?int.end, x-:!int.!int.end \vdash x+(z:int).x+(z:int).0 \mid x-!1.x-!2.0 \end{array} $	$ \begin{array}{c} x-:end \text{ completed} \\ \hline \text{[T-Nil]} \\ x-:end \vdash \emptyset \\ \hline \text{[T-Out]} \\ x-:!int.end \vdash x-!2.0 \\ \hline \text{[T-Par]} \\ x-:!int.!int.end \vdash x-!1.x-!2.0 \end{array} $	$ \begin{array}{c} \text{[T-Aux]} \\ \emptyset \vdash 2:int \\ \hline \text{[T-Aux]} \\ \emptyset \vdash 1:int \\ \hline \text{[T-Out]} \end{array} $
--	---	---

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

e)

$$\begin{array}{c}
\text{X} \\
x-:!int.end \vdash x-!true.0 \quad \emptyset \vdash 1:int \\
\hline
\text{[T-Out]} \\
x+:?int.?int.end \vdash x+(z:int).x+(z:int).0 \quad x-:!int.!int.end \vdash x-!1.x-!true.0 \\
\hline
\text{[T-Par]} \\
x+:?int.?int.end, x-:!int.!int.end \vdash x+(z:int).x+(z:int).0 \mid x-!1.x-!true.0
\end{array}$$

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 $P \equiv (vx:?int.?int.end)(x+(z:int).x+(z:int).0 \mid x-!1.z+!x-.0)$.

$$\begin{array}{c}
\text{z+:!(int.end).end, z-:end, w:end} \vdash P \quad \text{----- [T-Aux]} \\
\text{ } \vdash 2:\text{int} \\
\text{----- [T-Out]} \\
\text{z+:!(int.end).end, z-:end, w:int.end} \vdash P \mid w!2.0 \\
\text{----- [T-In]} \\
\text{z+:!(int.end).end, z-?(int.end).end} \vdash P \mid z-(w:int.end).w!2.0
\end{array}$$

Basta ver el juicio de tipado de $\text{z+:!(int.end).end, z-:end, w:end} \vdash P$.

$ \begin{array}{c} \text{z-:end, w:end, x+:end, y:int completed} \\ \text{----- [T-Nil]} \\ \text{z-:end, w:end, x+:end, y:int} \vdash 0 \\ \text{----- [T-In]} \\ \text{z-:end, w:end, x+:?int.end, y:int} \vdash x+(y:int).0 \\ \text{----- [T-In]} \\ \text{z-:end, w:end, x+:?int.?int.end} \vdash x+(y:int).x+(y:int).0 \\ \text{----- [T-Par]} \\ \text{z+:!(int.end).end, z-:end, w:end, x+:?int.?int.end, x-:!int.!int.end} \vdash x+(z:int).x+(z:int).0 \mid x-!1.z+!x-.0 \\ \text{----- [T-Res]} \\ \text{z+:!(int.end).end, z-:end, w:end} \vdash (vx:?int.?int.end)(x+(z:int).x+(z:int).0 \mid x-!1.z+!x-.0) \end{array} $	$ \begin{array}{c} \text{z+:end completed} \\ \text{----- [T-Nil]} \\ \text{z+:end} \vdash 0 \\ \text{----- [T-Out]} \\ \text{z+:!(int.end).end, x-:!int.end} \vdash z+!x-.0 \quad \text{----- [T-Aux]} \\ \text{ } \vdash 1:\text{int} \\ \text{----- [T-Out]} \\ \text{z+:!(int.end).end, x-:!int.!int.end} \vdash x-!1.z+!x-.0 \end{array} $
--	---

Bien tipado.

g)

$$(vx:?int.?int.end)(x+(z:int).x+(z:int).0 \mid x-!1.z+!x-.0) \mid z-(w:int.end).w-!2.0$$

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]
y-:!(!int.end).end, x+:?int.end, x-:!int.end ⊢ y-!x-.x+?(z:int).0      X
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 ⊢ (vx:?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)

<pre> 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] y-:!(!int.end).end, x+:?int.end, x-:!int.end ⊢ y-!x-.x+?(z:int).0 ----- 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 ⊢ (vx:?int.end)(y-!x-.x+?(z:int).0 y+?(z:!int.end).z!1.0) </pre>	<pre> y+:end, z:end completed ----- [T-Nil] ----- [T-Aux] y+:end, z:end ⊢ 0 ∅ ⊢ 1:int ----- [T-Out] y+:end, z:!int.end ⊢ z!1.0 ----- [T-In] y+:?(!int.end).end ⊢ y+?(z:!int.end).z!1.0 ----- [T-Par] </pre>
---	---

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
```

$P \equiv y \triangleright [\text{Done}:\emptyset, \text{Next}:y?(z:\text{int}).y!1.\emptyset]$

	----- [T-Aux]	x+:end, y:end, z:int completed	----- [T-Aux]
	$\emptyset \vdash 1:\text{int}$	$x+:end, y:end, z:\text{int} \vdash \emptyset$	
		----- [T-Out]	
x+:end, y:end completed	----- [T-Nil]	x+:end, y:!int.end, z:int $\vdash y!1.\emptyset$	----- [T-In]
	$x+:end, y:end \vdash \emptyset$	$x+:end, y:?int.!int.end \vdash y?(z:\text{int}).y!1.\emptyset$	----- [T-Branch]
		$x+:end, y:\&[\text{Done}:end, \text{Next}:?int.!int.end] \vdash y \triangleright [\text{Done}:\emptyset, \text{Next}:y?(z:\text{int}).y!1.\emptyset]$	----- [T-In]
		$x+:?(\&[\text{Done}:end, \text{Next}:?int.!int.end]).end \vdash x+(y:\&[\text{Done}:end, \text{Next}:?int.!int.end]).y \triangleright [\text{Done}:\emptyset, \text{Next}:y?(z:\text{int}).y!1.\emptyset]$	

b)

$(\forall x:\otimes[\text{Pr}:!int.?bool.end, \text{Co}:!int.!int.?bool.end]) (P \mid Q)$

$P = (x+\triangleleft \text{Pr}.x+!1.x+(z:bool).\emptyset)$
 $Q = (x\triangleright[\text{Pr}:x-(z:\text{int}).x-!true.\emptyset, \text{Co}:x-(z:\text{int}).x-(z:\text{int}).x-!true.\emptyset])$

$S = \otimes[\text{Pr}:!int.?bool.end, \text{Co}:!int.!int.?bool.end]$
 $T = \&[\text{Pr}:?int.!bool.end, \text{Co}:?int.?int.!bool.end]$

(1)	(2)
-----	-----
$x+:S \vdash P$	$x-:T \vdash Q$
----- [T-Par]	
$x+:S, x-:T \vdash P \mid Q$	
----- [T-Res]	
$(\forall x:S) (P \mid Q)$	

Rama (1)

	x+:end, z:bool completed	
	----- [T-Nil]	
	x+:end, z:bool $\vdash \emptyset$	
----- [T-Aux]	----- [T-In]	
$\emptyset \vdash 1:\text{int}$	$x+:?bool.\text{end} \vdash x+(z:bool).\emptyset$	
	----- [T-Out]	
$x+:!int.?bool.\text{end} \vdash x+!1.x+(z:bool).\emptyset$		
	----- [T-Choice]	
$x+:\oplus[\text{Pr}:!int.?bool.\text{end}, \text{Co}:!int.!int.?bool.\text{end}] \vdash x+\triangleleft\text{Pr}.x+!1.x+(z:bool).\emptyset$		

Rama (2)

		x-:end, z:int completed
		----- [T-Aux] ----- [T-Nil]
	x-:end, z:int completed	$\emptyset \vdash \text{true:bool}$ $x-:end, z:int \vdash \emptyset$
----- [T-Aux]	----- [T-Nil]	----- [T-Out]
$\emptyset \vdash \text{true:bool}$	$x-:end, z:int \vdash \emptyset$	$x-:!bool.\text{end}, z:int \vdash x-!true.\emptyset$
	----- [T-Out]	----- [T-In]
$x-:!bool.\text{end}, z:int \vdash x-!true.\emptyset$		$x-:?int.!bool.\text{end}, z:int \vdash x-(z:int).x-!true.\emptyset$
	----- [T-In]	----- [T-In]
$x-:?int.!bool.\text{end} \vdash x-(z:int).x-!true.\emptyset$	$x-:?int.?int.!bool.\text{end} \vdash x-(z:int).x-(z:int).x-!true.\emptyset$	
		----- [T-Branch]
$x-:\&[\text{Pr}:?int.!bool.\text{end}, \text{Co}:?int.?int.!bool.\text{end}] \vdash (x-\triangleright[\text{Pr}:x-(z:int).x-!true.\emptyset, \text{Co}:x-(z:int).x-(z:int).x-!true.\emptyset])$		