

1 Ejercicio I.15

Definir las compuertas **And** y **Or**.

$$\text{And} := \lambda x. \lambda y. \text{match}(x, w. \text{match}(y, a. \text{inl}(*), b. \text{inr}(*)), z. \text{inr}(*))$$

$$\text{Or} := \lambda x. \lambda y. \text{match}(x, a. \text{inr}(*), b. \text{match}(y, c. \text{inl}(*), d. \text{inr}(*)))$$

2 Ejercicio I.16

Dar el tipo de **And** y **Or**.

2.1 Típo de And

El tipo de **And** es: $\text{Bool} \rightarrow \text{Bool} \rightarrow \text{Bool}$, donde $\text{Bool} = \top \vee \top$. Tomé $\Gamma = x : \text{Bool}, y : \text{Bool}$.

$$\frac{\Gamma \vdash x : \top \vee \top \quad ax}{\Gamma \vdash y : \top \vee \top}
\frac{\Gamma, a : \top \vdash * : \top \quad \top_i}{\Gamma, a : \top \vdash \text{inl}(*) : \top \vee \top \quad \vee_{i_1}}
\frac{\Gamma, b : \top \vdash * : \top \quad \top_i}{\Gamma, b : \top \vdash \text{inr}(*) : \top \vee \top \quad \vee_{i_2}}
\frac{\Gamma, b : \top \vdash * : \top \quad \top_i}{\Gamma, z : \top \vdash * : \top \quad \vee_e}
\frac{\Gamma, z : \top \vdash * : \top \quad \top_i}{\Gamma, z : \top \vdash \text{inr}(*) : \text{Bool} \quad \vee_e}$$

$$\frac{\Gamma \vdash \text{match}(y, a.\text{inl}(*), b.\text{inr}(*)) : \text{Bool}}{\Gamma \vdash \text{match}(x, w.\text{match}(y, a.\text{inl}(*), b.\text{inr}(*)), z.\text{inr}(*) : \text{Bool}}$$

$$\frac{x : \text{Bool} \vdash \lambda y. \text{match}(x, w.\text{match}(y, a.\text{inl}(*), b.\text{inr}(*)), z.\text{inr}(*) : \text{Bool} \rightarrow \text{Bool}}{\vdash \lambda x. \lambda y. \text{match}(x, w.\text{match}(y, a.\text{inl}(*), b.\text{inr}(*)), z.\text{inr}(*) : \text{Bool} \rightarrow \text{Bool} \rightarrow \text{Bool} \rightarrow_i \text{Bool}}$$

2.2 Tipo de Or

El tipo de **Or** es: $\text{Bool} \rightarrow \text{Bool} \rightarrow \text{Bool}$, donde $\text{Bool} = \top \vee \perp$. Tomé $\Gamma = x : \text{Bool}, y : \text{Bool} \vee \Gamma_1 = \Gamma, b : \top$

$$\begin{array}{c}
\frac{}{\Gamma \vdash x : \top \vee \top} ax \quad \frac{\Gamma, a : \top \vdash * : \top}{\Gamma, a : \top \vdash \text{inr}(* : \top \vee \top)} \top_i \\
\hline
\Gamma \vdash \text{inr}(*) : \top \vee \top \quad \frac{\Gamma_1 \vdash y : \top \vee \top}{\Gamma_1, c : \top \vdash * : \top} ax \quad \frac{\Gamma_1, c : \top \vdash * : \top}{\Gamma_1, c : \top \vdash \text{inl}(* : \top \vee \top)} \top_i \\
\hline
\frac{}{\Gamma_1 \vdash \text{match}(x, a.\text{inr}(*), b.\text{match}(y, c.\text{inl}(*), d.\text{inr}(*))) : \text{Bool}} \top_i \\
\frac{}{\Gamma \vdash \text{match}(x, a.\text{inr}(*), b.\text{match}(y, c.\text{inl}(*), d.\text{inr}(*))) : \text{Bool}} \top_i \\
\frac{x : \text{Bool}, y : \text{Bool} \vdash \text{match}(x, a.\text{inr}(*), b.\text{match}(y, c.\text{inl}(*), d.\text{inr}(*))) : \text{Bool}}{x : \text{Bool} \vdash \lambda y. \text{match}(x, a.\text{inr}(*), b.\text{match}(y, c.\text{inl}(*), d.\text{inr}(*))) : \text{Bool} \rightarrow \text{Bool}} \rightarrow_i \\
\frac{}{\vdash \lambda x. \lambda y. \text{match}(x, a.\text{inr}(*), b.\text{match}(y, c.\text{inl}(*), d.\text{inr}(*))) : \text{Bool} \rightarrow \text{Bool} \rightarrow \text{Bool}} \rightarrow_i
\end{array}$$

3 Ejercicio II.5

Demostrar el teorema 4.12. En **Set**, los epimorfismos son las funciones sobreyectivas. Osea, toda flecha $f : A \rightarrow B$ cumple que $\forall b \in B, \exists a \in A$ tal que $f(a) = b$.

Ida: f es un epimorfismo $\implies f$ es sobreyectiva. Vamos a demostrarlo por el contrarrecíproco. Es decir f no es sobreyectiva $\implies f$ no es un epimorfismo. Entonces, quiero buscar $g : B \rightarrow C, h : B \rightarrow C$ con $g \neq h$ pero $g \circ f = h \circ f$.

La idea es que $g(b_0) \neq h(b_0)$ para algún $b_0 \in B$, pero que tengan el mismo comportamiento sobre la imagen de f .

Por simpleza tomemos $g(b) = 0$ y $h(b) = \begin{cases} 0 & \text{si } b \neq b_0 \\ 1 & \text{si } b = b_0 \end{cases}$

De este modo, $g(b_0) \neq h(b_0)$, por ende $g \neq h$.

Veamos ahora $g \circ f = h \circ f \iff (g \circ f)(a) = (h \circ f)(a) \forall a \in A$. Vemos que $f(a) \neq b_0 \forall a \in A$. Luego,

- $(g \circ f)(a) = g(f(a)) = 0$, pues g es constante y $f(a) \in Dom(g)$
- $(h \circ f)(a) = h(f(a)) = 0$, pues $f(a) \neq b_0$ por hipótesis.

Luego, vemos que $g \circ f = h \circ f$ para todo $a \in A$, pero no vale que $g = h$, por ende f no es un epimorfismo.

Vuelta: f es sobreyectiva $\implies f$ es un epimorfismo. Quiero ver que $g \circ f = h \circ f \implies g = h$, para $g : B \rightarrow C, h : B \rightarrow C$. Como asumo que f es sobreyectiva, sé que toda su imagen B está alcanzada por un $f(a)$, con $a \in A$.

Vamos a demostrarlo por absurdo. Asumamos que $g(b_1) \neq h(b_1)$ para algún $b_1 \in B$. Luego, como $b_1 \in B$, sabemos que existe un $a_1 \in A$ tal que $f(a_1) = b_1$.

Entonces, como sabemos que $g \circ f = h \circ f$, tenemos que:

$$\begin{aligned} (g \circ f)(a_1) &= (h \circ f)(a_1) \\ g(f(a_1)) &= h(f(a_1)) \\ g(b_1) &= h(b_1) \end{aligned}$$

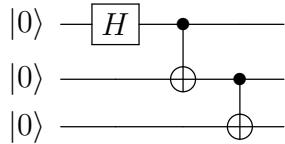
Absurdo! Pues habíamos asumido que $g(b_1) \neq h(b_1)$. Luego, $g = h$. Entonces, f es un epimorfismo.

□

4 Ejercicio III.15

Dar un circuito que genere el estado $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$ a partir de la entrada $|000\rangle$.

Un circuito que cumple es el siguiente:



Veamos que esto es correcto. Comenzamos con el estado $|\psi_0\rangle = |000\rangle$, que es lo mismo que $|0\rangle \otimes |0\rangle \otimes |0\rangle$. Vemos que al aplicar Hadamard al primer qubit obtenemos lo siguiente:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \otimes |0\rangle$$

Luego, al aplicar el CNOT con el control en el primer qubit y el target en el segundo, tenemos lo siguiente:

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$$

Esto es porque el control es el primer qubit. Entonces, si el primer qubit es 0, no hace nada (entonces queda en 0 el segundo). Pero si el primero es 1, se cambia el segundo qubit a 1.

Luego, al aplicar el CNOT con el control en el segundo qubit y el target en el tercero, sucede lo mismo pero en el tercer qubit.

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Que es lo mismo que $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$.

5 Ejercicio III.24

En el protocolo BB84, Alice y Bob descartan todos los bits en los que eligieron bases distintas, y se quedan con una clave candidata. Luego eligen al azar algunos bits de esa clave, los comparan públicamente y los descartan, para detectar la posible presencia de un espía (Eve). ¿Cuántos bits necesitan comparar Alice y Bob para tener al menos un 90% de probabilidad de detectar la presencia de Eve, suponiendo que Eve mide siempre en una base al azar?

En la comunicación, si Eve está espiando el canal y mide en una base al azar, va a afectar a lo que reciba Bob (va a medirlo y luego enviarle su medición a Bob). Ya que mide en una base al azar, va a medir correctamente o incorrectamente con la misma probabilidad ($\frac{1}{2}$ para cada caso). Luego, cuando Bob reciba esa medición, va a elegir una base al azar para medirlo. Entonces, a pesar de que no coincidan las bases de Bob y Eve, Bob tiene una probabilidad de $\frac{1}{2}$ de medir lo mismo que había enviado Alice. Esto es porque si difieren las bases, los valores de 0 y 1 (en cada base) estarían entrelazados, y Bob al elegir una base al azar, va a medir correctamente con probabilidad $\frac{1}{2}$.

Para detectar un error, Alice y Bob comparan algunos bits de su clave candidata (es decir, los bits en los que eligieron la misma base). Si hay alguna diferencia, significa que hay alguien espiando la comunicación, ya que si Alice envió un valor en una base, y Bob lo midió en esa base, deberían ser el mismo; la única opción para que no coincidan es que hay alguien alterando la comunicación. De todos modos, puede suceder que a pesar de que Eve haya medido en una base distinta a la de Alice, Bob obtenga el mismo valor que Alice envió. Asumimos que Bob y Alice miden en la misma base, ya que estamos hablando de **bits de la clave candidata**, que son los que Bob y Alice coinciden en la base.

Veamos la probabilidad de que se detecte un error en 1 bit. Sea A el evento en el que: "Eve mide en una base distinta a la de Alice" y B el evento en el que: "Bob no mide lo mismo que Alice envió".

$$P(\text{detectar error en 1 bit}) = P(A) \cdot P(B) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$$

Luego, la probabilidad de que no se detecte un error en 1 bit es $1 - \frac{1}{4} = \frac{3}{4}$.

Veamos que sucedería en n bits. Para que no se detecte un error en n bits, Eve tendría que pasar desapercibida en todos los bits. Luego, la probabilidad de que no se detecte un error en n bits es $(\frac{3}{4})^n$.

Lo que queremos es ver qué n necesitamos para que la probabilidad de **detectar un error** sea al menos 0.9

$$P(\text{detectar un error en } n \text{ bits}) = 1 - P(\text{no detectar un error en } n \text{ bits}) = 1 - \left(\frac{3}{4}\right)^n$$

Queremos que esta probabilidad sea al menos 0.9.

$$1 - \left(\frac{3}{4}\right)^n \geq 0.9$$

$$0.1 \geq \left(\frac{3}{4}\right)^n$$

$$\ln(0.1) \geq n \cdot \ln\left(\frac{3}{4}\right)$$

$$\frac{\ln(0.1)}{\ln\left(\frac{3}{4}\right)} \leq n$$

Dado que $\frac{\ln(0.1)}{\ln\left(\frac{3}{4}\right)} \approx 8.003$, con tomar $n = 9$ nos alcanza.

6 Algoritmo de codificación superdensa

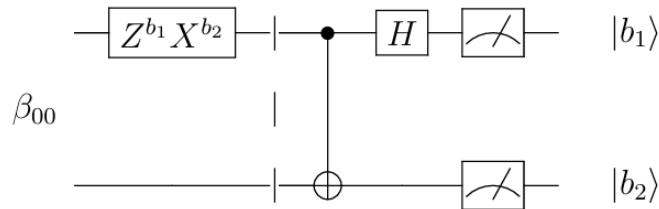
6.1 Traza del algoritmo para los distintos inputs

El algoritmo de codificación superdensa tiene como objetivo transmitir 2 bits clásicos enviando 1 qubit entre un emisor llamado Alice y un receptor llamado Bob.

Se parte de un estado inicial de bell $\beta_{00} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

Alice hace: $(Z^{b_1} X^{b_2} \otimes I)\beta_{00}$

Bob aplica la siguiente transformación al resultado de lo obtenido por Alice: $(H \otimes I)CNOT$ y realiza una medición.



6.1.1 Traza del algoritmo para bit 00

Parte Alice:

$$(Z^0 X^0 \otimes I)\beta_{00} = (I \otimes I)\beta_{00}$$

$$= (I \otimes I)\beta_{00} = \beta_{00}$$

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Parte Bob:

$$(H \otimes I)CNOT \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

$$= (H \otimes I) \frac{1}{\sqrt{2}}(CNOT|00\rangle + CNOT|11\rangle)$$

$$= (H \otimes I) \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$$

$$\begin{aligned}
&= \frac{1}{\sqrt{2}}((H \otimes I)|00\rangle + (H \otimes I)|10\rangle) \\
&= \frac{1}{\sqrt{2}}\left(\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes |0\rangle\right) + \left(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) \otimes |0\rangle \\
&= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) + \frac{1}{\sqrt{2}}(|00\rangle - |10\rangle)\right) \\
&= \frac{1}{2}(|00\rangle + |10\rangle + |00\rangle - |10\rangle) \\
&= \frac{1}{2}(2|00\rangle) = |00\rangle
\end{aligned}$$

6.2 Traza del algoritmo para bit 01

Parte Alice:

$$\begin{aligned}
(Z^0 X^1 \otimes I)\beta_{00} &= (X \otimes I)\beta_{00} \\
&= (X \otimes I)\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\
&= \frac{1}{\sqrt{2}}((X \otimes I)|00\rangle + (X \otimes I)|11\rangle) \\
&= \frac{1}{\sqrt{2}}((X|0\rangle) \otimes (I|0\rangle) + (X|1\rangle) \otimes (I|1\rangle)) \\
&= \frac{1}{\sqrt{2}}(|1\rangle \otimes |0\rangle + |0\rangle \otimes |1\rangle) \\
&= \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle) = \beta_{01}
\end{aligned}$$

Parte Bob:

$$\begin{aligned}
(H \otimes I)CNOT\beta_{01} &= (H \otimes I)(CNOT\frac{1}{\sqrt{2}}(|10\rangle + |01\rangle)) \\
&= (H \otimes I)\left(\frac{1}{\sqrt{2}}(CNOT|10\rangle + CNOT|01\rangle)\right) \\
&= (H \otimes I)\left(\frac{1}{\sqrt{2}}(|11\rangle + |01\rangle)\right) \\
&= \frac{1}{\sqrt{2}}((H \otimes I)|11\rangle + (H \otimes I)|01\rangle) \\
&= \frac{1}{\sqrt{2}}(H|1\rangle \otimes I|1\rangle + H|0\rangle \otimes I|1\rangle) \\
&= \frac{1}{\sqrt{2}}\left(((\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)) \otimes |1\rangle) + (\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) \otimes |1\rangle\right) \\
&= \frac{1}{\sqrt{2}}\left(((\frac{1}{\sqrt{2}}(|01\rangle - |11\rangle)) + (\frac{1}{\sqrt{2}}(|01\rangle + |11\rangle)))\right) \\
&= \frac{1}{\sqrt{2}}(\frac{1}{\sqrt{2}}(|01\rangle - |11\rangle + |01\rangle + |11\rangle)) \\
&= \frac{1}{2}(2|01\rangle) \\
&= |01\rangle
\end{aligned}$$

6.3 Traza del algoritmo para bit 10

Parte Alice:

$$\begin{aligned}
(Z^1 X^0 \otimes I)\beta_{00} &= (Z \otimes I)\beta_{00} \\
&= (Z \otimes I)\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\
&= \frac{1}{\sqrt{2}}((Z \otimes I)|00\rangle + (Z \otimes I)|11\rangle) \\
&= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = \beta_{10}
\end{aligned}$$

Parte Bob:

$$\begin{aligned}
(H \otimes I)CNOT\beta_{10} &= (H \otimes I)(CNOT\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle))
\end{aligned}$$

$$\begin{aligned}
&= (H \otimes I) \left(\frac{1}{\sqrt{2}} (CNOT |00\rangle - CNOT |11\rangle) \right) \\
&= (H \otimes I) \left(\frac{1}{\sqrt{2}} (|00\rangle - |10\rangle) \right) \\
&= \frac{1}{\sqrt{2}} ((H \otimes I) |00\rangle - (H \otimes I) |10\rangle) \\
&= \frac{1}{\sqrt{2}} (H |0\rangle \otimes I |0\rangle - H |1\rangle \otimes I |0\rangle) \\
&= \frac{1}{\sqrt{2}} (((\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)) \otimes |0\rangle) - (\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)) \otimes |0\rangle) \\
&= \frac{1}{\sqrt{2}} (((\frac{1}{\sqrt{2}} (|00\rangle + |10\rangle))) - (\frac{1}{\sqrt{2}} (|00\rangle - |10\rangle))) \\
&= \frac{1}{\sqrt{2}} (\frac{1}{\sqrt{2}} (|00\rangle + |10\rangle - |00\rangle + |10\rangle)) \\
&= \frac{1}{2} (2 |10\rangle) \\
&= |10\rangle
\end{aligned}$$

6.4 Traza del algoritmo para bit 11

Parte Alice:

$$\begin{aligned}
&(Z^1 X^1 \otimes I) \beta_{00} = (ZX \otimes I) \beta_{00} \\
&= (ZX \otimes II) \beta_{00} \\
&= (Z \otimes I)((X \otimes I) \beta_{00}) \\
&= (Z \otimes I)((X \otimes I) \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)) \\
&= (Z \otimes I)(\frac{1}{\sqrt{2}} ((X \otimes I) |00\rangle + (X \otimes I) |11\rangle)) \\
&= (Z \otimes I)(\frac{1}{\sqrt{2}} ((X |0\rangle) \otimes (I |0\rangle) + (X |1\rangle) \otimes (I |1\rangle))) \\
&= (Z \otimes I)(\frac{1}{\sqrt{2}} (|1\rangle \otimes |0\rangle + |0\rangle \otimes |1\rangle)) \\
&= (Z \otimes I)(\frac{1}{\sqrt{2}} (|10\rangle + |01\rangle)) \\
&= \frac{1}{\sqrt{2}} ((Z \otimes I) |10\rangle + (Z \otimes I) |01\rangle) \\
&= \frac{1}{\sqrt{2}} (-|10\rangle + |01\rangle) = \beta_{11}
\end{aligned}$$

Parte Bob:

$$\begin{aligned}
&(H \otimes I) CNOT \beta_{11} \\
&= (H \otimes I) (CNOT \frac{1}{\sqrt{2}} (|01\rangle - |10\rangle)) \\
&= (H \otimes I) \left(\frac{1}{\sqrt{2}} (CNOT |01\rangle - CNOT |10\rangle) \right) \\
&= (H \otimes I) \left(\frac{1}{\sqrt{2}} (|01\rangle - |11\rangle) \right) \\
&= \frac{1}{\sqrt{2}} ((H \otimes I) |01\rangle - (H \otimes I) |11\rangle) \\
&= \frac{1}{\sqrt{2}} (H |0\rangle \otimes I |1\rangle - H |1\rangle \otimes I |1\rangle) \\
&= \frac{1}{\sqrt{2}} (((\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)) \otimes |1\rangle) - (\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)) \otimes |1\rangle) \\
&= \frac{1}{\sqrt{2}} (((\frac{1}{\sqrt{2}} (|01\rangle + |11\rangle))) - (\frac{1}{\sqrt{2}} (|01\rangle - |11\rangle))) \\
&= \frac{1}{\sqrt{2}} (\frac{1}{\sqrt{2}} (|01\rangle + |11\rangle - |01\rangle + |11\rangle)) \\
&= \frac{1}{2} (2 |11\rangle) \\
&= |11\rangle \\
\langle 11 | |11\rangle^2 &= 1 \implies \text{la probabilidad de caer en el estado } |11\rangle \text{ es del 100\%}
\end{aligned}$$

6.5 Algoritmo de cálculo lambda para computación cuántica con control clásico

Sean las siguientes definiciones:

1. **EPR** : $!(\top \multimap (\text{qbit} \otimes \text{qbit}))$
 $\text{EPR} = \lambda x. \text{CNOT}\langle H(\text{new } 0), \text{new } 0 \rangle$
2. **U** : $!(\text{qbit} \multimap (\text{bit} \otimes \text{bit} \multimap \text{qbit}))$
 $U = \lambda q. \lambda \langle x, y \rangle. \text{if } x \text{ then } (\text{if } y \text{ then } U_{11}q \text{ else } U_{10}q) \\ \text{else } (\text{if } y \text{ then } U_{01}q \text{ else } U_{00}q)$

donde:

$$U_{00} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad U_{01} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad U_{10} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad U_{11} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

Algoritmo:

Condensacion :!bit \multimap !bit \multimap !(bit \otimes bit)

```
Condensacion =  $\lambda b_1 \lambda b_2. (\text{let } \langle x, y \rangle = \text{EPR}^* \text{ in}$ 
 $\text{let } a = Ux \langle b_1, b_2 \rangle \text{ in}$ 
 $\text{let } \langle q_1, q_2 \rangle = \text{CNOT}\langle a, y \rangle \text{ in } \langle \text{meas}(Hq_1), \text{meas } q_2 \rangle \rangle$ 
```

7 Tipado de la función de condensación

7.1 Derivación del tipo

Para el tipado de la función de condensación, asumimos $\Gamma = \{b_1 : !\text{bit}, b_2 : !\text{bit}\}$. El objetivo es derivar:

$$\vdash \text{Condensacion} : !\text{bit} \multimap_{\circ} !(\text{bit} \otimes \text{bit})$$

Definimos los siguientes términos para simplificar la notación:

- $M_1 = \text{EPR} *$
- $M_2 = U x \langle b_1, b_2 \rangle$
- $M_3 = \text{CNOT} \langle a, y \rangle$
- $M_4 = \langle \text{meas}(H q_1), \text{meas } q_2 \rangle$

Estructura general:

$$\frac{\begin{array}{c} (1) \\ \vdash M_1 : \text{qbit} \otimes \text{qbit} \end{array} \quad \begin{array}{c} (2) \\ \Gamma, x : \text{qbit}, y : \text{qbit} \vdash \text{let } a = M_2 \text{ in let } \langle q_1, q_2 \rangle = M_3 \text{ in } M_4 : !(\text{bit} \otimes \text{bit}) \end{array}}{\frac{\Gamma \vdash \text{let } \langle x, y \rangle = M_1 \text{ in } \dots : !(\text{bit} \otimes \text{bit})}{\frac{\begin{array}{c} b_1 : !\text{bit} \vdash \lambda b_2. \dots : !\text{bit} \multimap_{\circ} !(\text{bit} \otimes \text{bit}) \\ \vdash \lambda b_1 \lambda b_2. \dots : !\text{bit} \multimap_{\circ} !(\text{bit} \otimes \text{bit}) \end{array}}{\vdash \lambda b_1 \lambda b_2. \dots : !\text{bit} \multimap_{\circ} !(\text{bit} \otimes \text{bit})}} \multimap_I} \otimes_E$$

(1) Derivación de M_1 :

$$\frac{\vdash \text{EPR} : !(\top \multimap_{\circ} \text{qbit} \otimes \text{qbit}) \quad ax}{\vdash \text{EPR} : \top \multimap_{\circ} \text{qbit} \otimes \text{qbit}} \frac{sub}{\vdash * : \top} \frac{ax}{\vdash \text{EPR} * : \text{qbit} \otimes \text{qbit}} \multimap_E$$

(2) Parte principal: Separamos la derivación del cuerpo. Sea $N = \text{let } \langle q_1, q_2 \rangle = M_3 \text{ in } M_4$. Usamos la regla de aplicación para el ‘let’ a = ... (equivalente a $(\lambda a.N)M_2$).

$$\frac{}{\Gamma, x : \text{qbit} \vdash M_2 : \text{qbit}} \quad \frac{(2.1)}{\Gamma, x : \text{qbit}, y : \text{qbit} \vdash \text{let } a = M_2 \text{ in } N \text{ !}(bit \otimes bit)} \quad (2.2)$$

(2.1) Derivación de M_2 :

$$\frac{}{\vdash U \text{ !}(qbit \multimap (bit \otimes bit \multimap qbit))} \quad ax$$

$$\frac{}{\vdash U : \text{qbit} \multimap (bit \otimes bit \multimap qbit)} \quad sub$$

$$\frac{x : \text{qbit} \vdash x : \text{qbit} \quad var}{x : \text{qbit} \vdash x : \text{qbit}} \quad o_E$$

$$\frac{\frac{b_1 : \text{!bit} \vdash b_1 : \text{!bit} \quad var}{b_1 : \text{!bit} \vdash b_1 : \text{bit}} \quad sub \quad \frac{b_2 : \text{!bit} \vdash b_2 : \text{!bit} \quad var}{b_2 : \text{!bit} \vdash b_2 : \text{bit}} \quad sub}{b_1 : \text{!bit} \vdash b_1 : \text{bit} \quad \frac{b_2 : \text{!bit} \vdash b_2 : \text{bit}}{\Gamma \vdash \langle b_1, b_2 \rangle : \text{bit} \otimes \text{bit}}} \quad \otimes_I$$

$$\Gamma, x : \text{qbit} \vdash U x \langle b_1, b_2 \rangle : \text{qbit} \quad o_E$$

(2.2) Derivación de N : Recordamos $N = \text{let } \langle q_1, q_2 \rangle = \text{CNOT}\langle a, y \rangle \text{ in } M_4$.

$$\frac{}{\vdash \text{CNOT !}(qbit^2 \multimap qbit^2) \quad ax} \quad (3)$$

$$\frac{}{\vdash \text{CNOT : qbit}^2 \multimap \text{qbit}^2 \quad sub}$$

$$\frac{a : \text{qbit}, y : \text{qbit} \vdash \langle a, y \rangle : \text{qbit}^2 \quad o_E}{a : \text{qbit}, y : \text{qbit} \vdash \text{CNOT}\langle a, y \rangle : \text{qbit} \otimes \text{qbit}}$$

$$\frac{a : \text{qbit}, y : \text{qbit} \vdash \text{let } \langle q_1, q_2 \rangle = \dots \text{ in } M_4 \text{ !}(bit} \otimes \text{bit) \quad \otimes_E}{q_1 : \text{qbit}, q_2 : \text{qbit} \vdash M_4 \text{ !}(bit} \otimes \text{bit)}$$

(3) Derivación de M_4 : $M_4 = \langle \text{meas}(H q_1), \text{meas} q_2 \rangle$. Usamos la regla \otimes_I con $n = 1$.

$$\frac{}{\vdash \text{meas} : (\text{qbit} \multimap \text{!bit}) \quad ax} \quad \frac{\vdash H : !(\text{qbit} \multimap \text{qbit}) \quad ax}{\vdash H : !(\text{qbit} \multimap \text{qbit}) \quad ax}$$

$$\frac{}{\vdash \text{meas} : \text{!o!bit}} \quad sub \quad \frac{}{\vdash H : \text{qbit} \multimap \text{qbit}} \quad sub$$

$$\frac{q_1 : \text{qbit} \vdash q_1 : \text{qbit} \quad q_1 : \text{qbit} \vdash q_1 : \text{qbit}}{\vdash \text{meas} : \text{qbit} \multimap \text{!bit} \quad o_E}$$

$$\frac{q_1 : \text{qbit} \vdash \langle \text{meas}(H q_1), \text{meas} q_2 \rangle : !(\text{bit} \otimes \text{bit}) \quad q_2 : \text{qbit} \vdash q_2 : \text{qbit} \quad o_E}{q_1 : \text{qbit} \vdash \langle \text{meas}(H q_1), \text{meas} q_2 \rangle : !(\text{bit} \otimes \text{bit}) \quad \otimes_I(n=1)}$$