1 Ejercicio I.15

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

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

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

2 Ejercicio I.16

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

2.1 Tipo de And

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

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

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

$$\dfrac{x : \text{Bool} \vdash \lambda y.match(x, w.match(y, a.inl(*), b.inr(*)), z.inr(*)) : \text{Bool} \to \text{Bool}}{\vdash \lambda x.\lambda y.match(x, w.match(y, a.inl(*), b.inr(*)), z.inr(*)) : \text{Bool} \to \text{Bool} \to \text{Bool}}\, \to_i$$

2.2 Tipo de Or

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

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash x : \top \vee \top}\, ax \quad \dfrac{\dfrac{}{\Gamma, a : \top \vdash * : \top}\, \top_i}{\Gamma, a : \top \vdash inr(*) : \top \vee \top}\, \vee_{i_2} \quad \dfrac{}{\Gamma_1 \vdash y : \top \vee \top}\, ax \quad \dfrac{\dfrac{}{\Gamma_1, c : \top \vdash * : \top}\, \top_i}{\Gamma_1, c : \top \vdash inl(*) : \top \vee \top}\, \vee_{i_1} \quad \dfrac{\dfrac{}{\Gamma_1, d : \top \vdash * : \top}\, \top_i}{\Gamma_1, d : \top \vdash inr(*) : \top \vee \top}\, \vee_{i_2}}{\Gamma_1 \vdash match(y, c.inl(*), d.inr(*)) : \text{Bool}}\, \vee_e}{\Gamma \vdash match(x, a.inr(*), b.match(y, c.inl(*), d.inr(*))) : \text{Bool}}\, \vee_e}{x : \text{Bool}, y : \text{Bool} \vdash match(x, a.inr(*), b.match(y, c.inl(*), d.inr(*))) : \text{Bool}}\, \to_i}{x : \text{Bool} \vdash \lambda y.match(x, a.inr(*), b.match(y, c.inl(*), d.inr(*))) : \text{Bool} \to \text{Bool}}\, \to_i}{\vdash \lambda x.\lambda y.match(x, a.inr(*), b.match(y, c.inl(*), d.inr(*))) : \text{Bool} \to \text{Bool} \to \text{Bool}}\, \to_i$$

3 Ejercicio II.5

Demostrar el teorema 4.12. En **Set**, los epimorfismos son las funciones sobreyectivas. Osea, toda flecha $f : A \to B$ 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 $g : B \to C, h : B \to C$ 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 $g : B \to C, h : B \to C$. 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:

$$(g \circ f)(a_1) = (h \circ f)(a_1)$$
$$g(f(a_1)) = h(f(a_1))$$
$$g(b_1) = h(b_1)$$

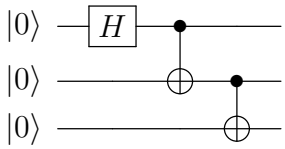
Absurdo! Pues habíamos asumido que $g(b_1) \neq h(b_1)$. Luego, $g = h$. Entonces, f es un epimorfismo.

□

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

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

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.