Modelos de computación cuánticos

Doble Grado en Ingeniería Informática y Matemáticas

Pablo Baeyens Fernández

25 de Junio de 2019

Trabajo Fin de Grado

E.T.S. de Ingenierías Informática y de Telecomunicación
Facultad de Ciencias

UNIVERSIDAD DE GRANADA

Índice

Introducción

La computación cuántica supone el uso de un nuevo tipo de ordenador basado en los principios de la física cuántica que aprovecha sus propiedades para conseguir ventajas asintóticas respecto de los mejores algoritmos clásicos.

La teoría de la complejidad nos permite estudiar sus propiedades de forma teórica y relacionarla con el modelo clásico.

Podemos además simular y verificar los algoritmos cuánticos con un ordenador clásico, asumiendo en su ejecución un coste de tiempo exponencial.

Teoría de la complejidad

Problemas promesa

Trabajamos con el alfabeto binario, $\mathbb{B} = \{0, 1\}$.

El punto de partida de la teoría de la computabilidad y la complejidad es el concepto de lenguaje: un subconjunto $L \subseteq \mathbb{B}^*$. A través de este podemos definir los problemas de decisión:

Definición

Un problema promesa *es un par de lenguajes disjuntos* $(L_{\text{Sí}}, L_{\text{No}})$.

El problema asociado consiste en clasificar una palabra $w \in L_{\text{Sí}} \cup L_{\text{No}}$ *como una instancia afirmativa o negativa.*

Si un problema promesa es una partición de $\mathbb{B}^*$ lo identificamos con $L_{\text{Sí}}$.

El enfoque clásico de la computabilidad y complejidad pasa por el uso de la máquina de Turing para definir el concepto de decidibilidad.

Con esta definimos:

Definición

Sea $L \subseteq \mathbb{B}^*$. Entonces

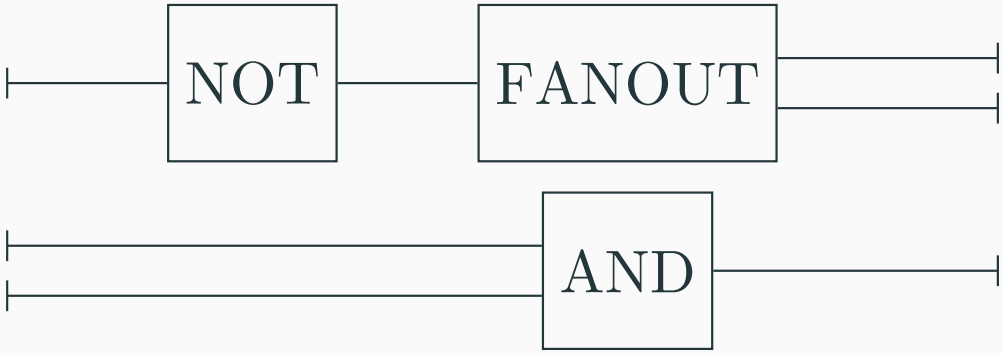
- $L \in$ P si es decidible en tiempo polinómico,
- $L \in$ NP si es *verificable* en tiempo polinómico y
- $L \in$ PSPACE si es decidible en espacio polinómico.

P $\subseteq$ NP $\subseteq$ PSPACE.

Permanece abierto el problema P $\stackrel{?}{=}$ PSPACE.

Circuitos clásicos

Los circuitos componen operaciones básicas de la forma $f : \mathbb{B}^n \to \mathbb{B}^m$.



Tienen una función asociada; en este caso $(\mathrm{FANOUT} \circ \mathrm{NOT}) \times \mathrm{AND}$.

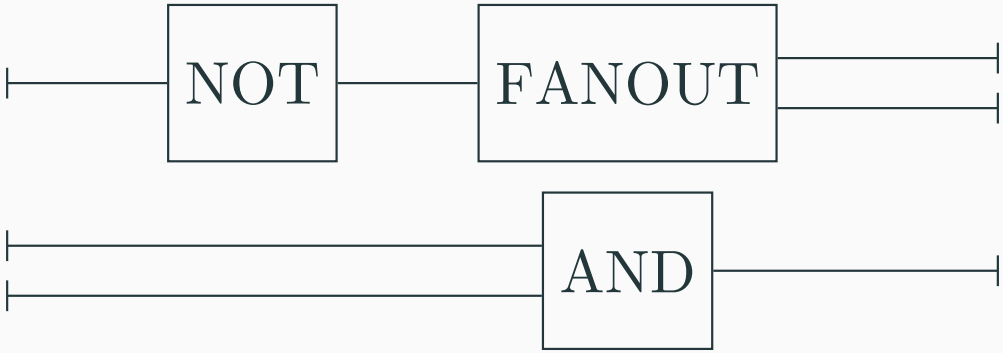
Definición

Una *familia de circuitos* es una sucesión de circuitos $\mathcal{C} = \{C_n\}$ tal que C_n tiene n entradas.

Es *uniforme* si la función $n \mapsto C_n$ es computable. $\mathcal{C}(w) = C_{|w|}(w)$.

Circuitos clásicos

Los circuitos componen operaciones básicas de la forma $f : \mathbb{B}^n \to \mathbb{B}^m$.



Tienen una función asociada; en este caso $(\mathrm{FANOUT} \circ \mathrm{NOT}) \times \mathrm{AND}$.

Definición

Una *familia de circuitos* es una sucesión de circuitos $\mathcal{C} = \{C_n\}$ tal que C_n tiene n entradas.

Es *uniforme* si la función $n \mapsto C_n$ es computable. $\mathcal{C}(w) = C_{|w|}(w)$.

P incluye los problemas decidibles en tiempo polinómico; se considera que tienen un algoritmo eficiente. Podemos caracterizarla con circuitos:

Proposición

$L \in \mathsf{P}$ *si y sólo si existe una familia* $\mathcal{C} = \{C_n\}$ *tal que*

- $n \mapsto C_n$ *es calculable en tiempo polinómico y*
- $$\mathcal{C}(w) = 1_L(w) = \begin{cases} 1 & \text{si } w \in L, \\ 0 & \text{en otro caso.} \end{cases} \qquad \forall w \in \mathbb{B}^*$$

Podemos caracterizar de forma análoga la clase NP.

Circuitos probabilísticos

Son un modelo intermedio entre el clásico y el cuántico.

Espacio de estados

Tomamos $R = \mathrm{Lin}_{\mathbb{R}}(|0\rangle, |1\rangle)$ y combinamos con el producto tensor $\otimes$.

Si $v \in R^{\otimes n}$ tiene todas sus entradas no negativas y $\|v\|_1 = 1$, representa una distribución de probabilidad sobre la base de $R^{\otimes n}$.

Las operaciones son aplicaciones lineales $f : R^{\otimes n} \to R^{\otimes m}$ con matrices estocásticas que se combinan en circuitos de forma análoga.

En este caso la operación es $\mathrm{NOT} \otimes (\mathrm{FANOUT} \circ \mathrm{RAND}\left(\frac{1}{2}\right))$.

Circuitos probabilísticos

Son un modelo intermedio entre el clásico y el cuántico.

Espacio de estados

Tomamos $R = \mathrm{Lin}_{\mathbb{R}}(|0\rangle, |1\rangle)$ y combinamos con el producto tensor $\otimes$.

Si $v \in R^{\otimes n}$ tiene todas sus entradas no negativas y $\|v\|_1 = 1$, representa una distribución de probabilidad sobre la base de $R^{\otimes n}$.

Las operaciones son aplicaciones lineales $f : R^{\otimes n} \to R^{\otimes m}$ con matrices estocásticas que se combinan en circuitos de forma análoga.

En este caso la operación es $\mathrm{NOT} \otimes (\mathrm{FANOUT} \circ \mathrm{RAND}\left(\frac{1}{2}\right))$.

Circuitos probabilísticos

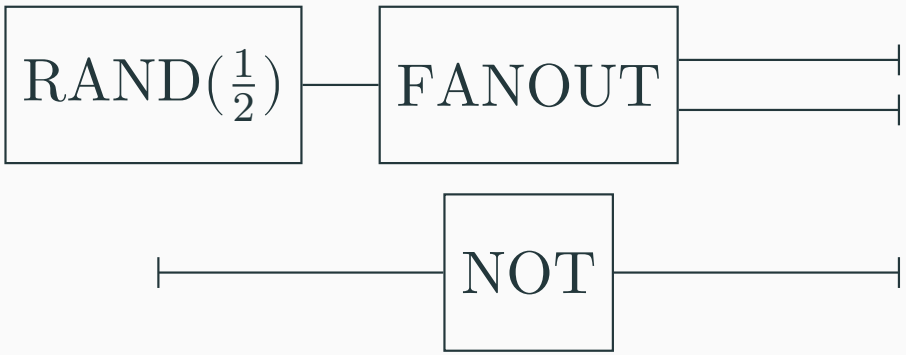
Son un modelo intermedio entre el clásico y el cuántico.

Espacio de estados

Tomamos $R = \mathrm{Lin}_{\mathbb{R}}(|0\rangle, |1\rangle)$ y combinamos con el producto tensor $\otimes$.

Si $v \in R^{\otimes n}$ tiene todas sus entradas no negativas y $\|v\|_1 = 1$, representa una distribución de probabilidad sobre la base de $R^{\otimes n}$.

Las operaciones son aplicaciones lineales $f : R^{\otimes n} \to R^{\otimes m}$ con matrices estocásticas que se combinan en circuitos de forma análoga.



En este caso la operación es $\mathrm{NOT} \otimes \left(\mathrm{FANOUT} \circ \mathrm{RAND}\left(\frac{1}{2}\right)\right)$.

Un circuito probabilístico tiene una variable aleatoria asociada.

Medición

Dada $w \in \mathbb{B}^*$ y un circuito C, si $(p_1, \dots, p_N) = C(w)$ notamos también por $C(w)$ la variable aleatoria tal que $P[C(|w\rangle) = |j\rangle] = p_j$.

Definición

$\mathcal{C}$ *calcula* $f : \mathbb{B}^* \to \mathbb{B}^*$ si $P[\mathcal{C}(w) = f(w)] > \frac{1}{2} \quad \forall w \in \mathbb{B}^*$

Definición

$\mathcal{C}$ *calcula* f *con error acotado* si $P[\mathcal{C}(w) = f(w)] > \frac{2}{3} \quad \forall w \in \mathbb{B}^*$

Un circuito probabilístico tiene una variable aleatoria asociada.

Medición

Dada $w \in \mathbb{B}^*$ y un circuito C, si $(p_1, \dots, p_N) = C(w)$ notamos también por $C(w)$ la variable aleatoria tal que $P[C(|w\rangle) = |j\rangle] = p_j$.

Definición

$\mathcal{C}$ *calcula* $f : \mathbb{B}^* \to \mathbb{B}^*$ si $P[\mathcal{C}(w) = f(w)] > \frac{1}{2} \quad \forall w \in \mathbb{B}^*$

Definición

$\mathcal{C}$ *calcula* f *con error acotado* si $P[\mathcal{C}(w) = f(w)] > \frac{2}{3} \quad \forall w \in \mathbb{B}^*$

Definición

Sea $\mathcal{C} = \{C_n\}$ una familia de circuitos probabilísticos calculable en tiempo polinómico y L un lenguaje. Decimos que

- $L \in$ PP si 1_L es calculable por $\mathcal{C}$ y
- $L \in$ BPP si lo es con error acotado.

Podemos relajar la cota del error acotado mediante las *cotas de Chernoff*.

Si el error no está acotado, el algoritmo inducido no es eficiente.

Proposición

NP $\subseteq$ PP

Definición

Sea $\mathcal{C} = \{C_n\}$ una familia de circuitos probabilísticos calculable en tiempo polinómico y L un lenguaje. Decimos que

- $L \in$ PP si 1_L es calculable por $\mathcal{C}$ y
- $L \in$ BPP si lo es con error acotado.

Podemos relajar la cota del error acotado mediante las *cotas de Chernoff*.

Si el error no está acotado, el algoritmo inducido no es eficiente.

Proposición

NP $\subseteq$ PP

Espacio de estados

Es el espacio proyectivo de un espacio de Hilbert complejo de dimensión finita. Tomamos $Q = \mathrm{Lin}_{\mathbb{C}}(|0\rangle, |1\rangle)$ (*qubit*) y combinamos con el producto tensor $\otimes$.

Un estado se identifica con un vector $|\psi\rangle \in Q^{\otimes n}$ con $\||\psi\rangle\|_2 = 1$.

Operaciones

Son aplicaciones unitarias

$$U : Q^{\otimes n} \to Q^{\otimes m},$$ con n entradas y m salidas.

Las tomamos de un *conjunto universal finito* como $\{\mathrm{CNOT}, H, R_{\pi/4}\}$.

Espacio de estados

Es el espacio proyectivo de un espacio de Hilbert complejo de dimensión finita. Tomamos $Q = \mathrm{Lin}_{\mathbb{C}}(|0\rangle, |1\rangle)$ (*qubit*) y combinamos con el producto tensor $\otimes$.

Un estado se identifica con un vector $|\psi\rangle \in Q^{\otimes n}$ con $\||\psi\rangle\|_2 = 1$.

Operaciones

Son aplicaciones unitarias

$$U : Q^{\otimes n} \to Q^{\otimes m},$$ con n entradas y m salidas.

Las tomamos de un *conjunto universal finito* como $\{\mathrm{CNOT}, H, R_{\pi/4}\}$.

Espacio de estados

Es el espacio proyectivo de un espacio de Hilbert complejo de dimensión finita. Tomamos $Q = \mathrm{Lin}_{\mathbb{C}}(|0\rangle, |1\rangle)$ (*qubit*) y combinamos con el producto tensor $\otimes$.

Un estado se identifica con un vector $|\psi\rangle \in Q^{\otimes n}$ con $\||\psi\rangle\|_2 = 1$.

Operaciones

Son aplicaciones unitarias

$$U : Q^{\otimes n} \to Q^{\otimes m}, \text{ con } n \text{ entradas y } m \text{ salidas.}$$

Las tomamos de un *conjunto universal finito* como $\{\mathrm{CNOT}, H, R_{\pi/4}\}$.

Para transformar estados cuánticos en estados clásicos, utilizamos una operación de medición.

Medición

Dada $w \in \mathbb{B}^*$ y un circuito cuántico C, si $(\alpha_1, \dots, \alpha_N) = C(|w\rangle)$ notamos también por $C(|w\rangle)$ la variable aleatoria tal que

$$P[C(|w\rangle) = |j\rangle] = |\alpha_j|^2.$$

La calculabilidad se define de forma análoga al caso probabilístico.

Los análogos a BPP y PP en el caso cuántico son BQP y PQP.

También NP tiene análogo en QMA.

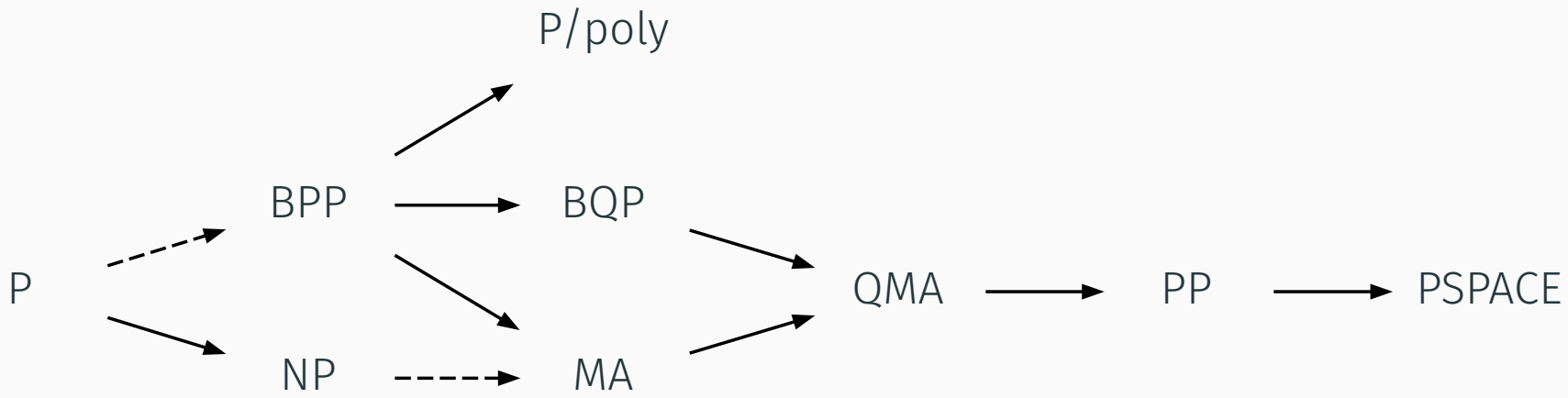
Proposición

Se tiene que

- PP $=$ PQP *y*
- QMA $\subseteq$ PP.

Jerarquía de clases

En el trabajo se demuestra la siguiente jerarquía de clases de complejidad:



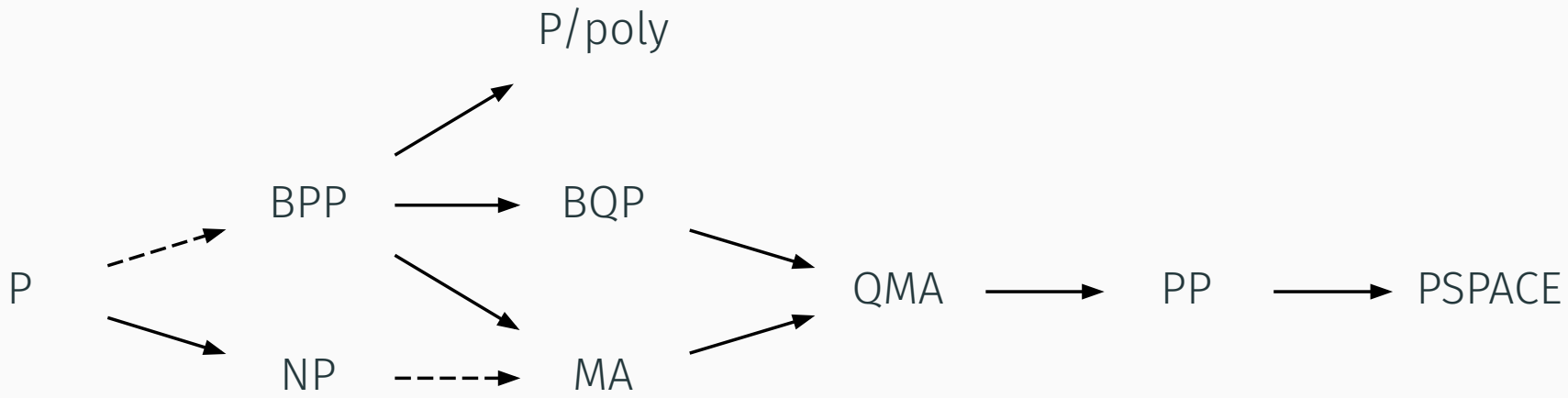
La cadena

$$\mathrm{BQP} \subseteq \mathrm{QMA} \subseteq \mathrm{PP}$$

nos da la mejor cota superior probabilística conocida de BQP.

Jerarquía de clases

En el trabajo se demuestra la siguiente jerarquía de clases de complejidad:



La cadena

$$\mathrm{BQP} \subseteq \mathrm{QMA} \subseteq \mathrm{PP}$$

nos da la mejor cota superior probabilística conocida de BQP.

Algoritmos cuánticos

Quipper

- Quipper es un lenguaje de programación funcional pura embebido en Haskell.
- Permite definir familias uniformes de circuitos mixtos, que combinan partes clásicas y cuánticas.
- Para definir los circuitos operamos en una mónada o usamos metaprogramación.

Oráculos

El punto de partida de muchos algoritmos cuánticos es el uso de oráculos: circuitos que calculan una cierta función reversible.

Un oráculo tiene una forma de entrada `shape` y un circuito `circuit`.

```
data Oracle qa = Oracle {
  shape   :: qa,
  circuit :: (qa,Qubit) → Circ (qa,Qubit)
}
```

Construyo los oráculos a partir de tablas de verdad usando metaprogramación.

Oráculos

El punto de partida de muchos algoritmos cuánticos es el uso de oráculos: circuitos que calculan una cierta función reversible.

Un oráculo tiene una forma de entrada `shape` y un circuito `circuit`.

```
data Oracle qa = Oracle {
  shape   :: qa,
  circuit :: (qa,Qubit) → Circ (qa,Qubit)
}
```

Construyo los oráculos a partir de tablas de verdad usando metaprogramación.

Ejemplo de oráculo

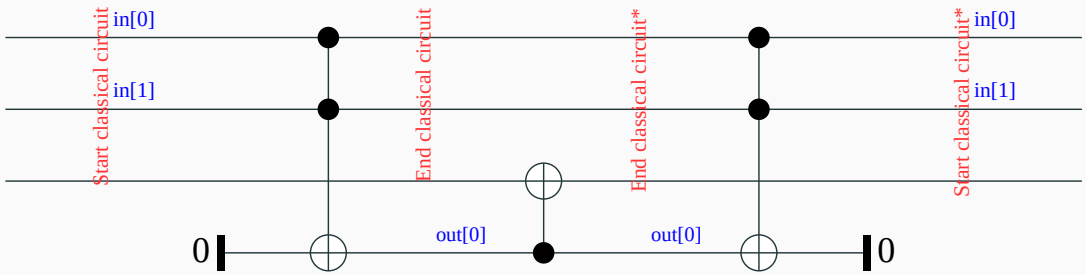
Dada $f : \mathbb{B}^n \to \mathbb{B}^m$, la transformamos en $g : \mathbb{B}^n \times \mathbb{B}^m \to \mathbb{B}^n \times \mathbb{B}^m$ dada por

$$g(x, y) = (x, y \oplus f(x)),$$

donde $\oplus$ es la operación XOR bit a bit.

Transformo una tabla de verdad en un circuito que calcula de forma reversible esa función.

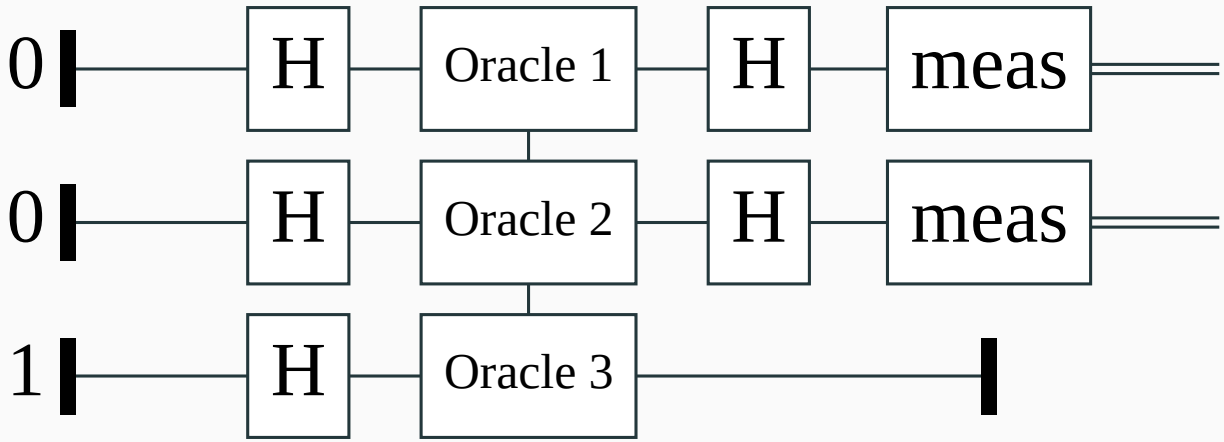
```
00,0
01,0
10,0
11,1
```



Algoritmo de Deutsch-Jozsa

Determina en una consulta si un predicado es constante o balanceado.

```
deutschJozsa :: (QShape ba qa ca) => Oracle qa → Circ ca
```

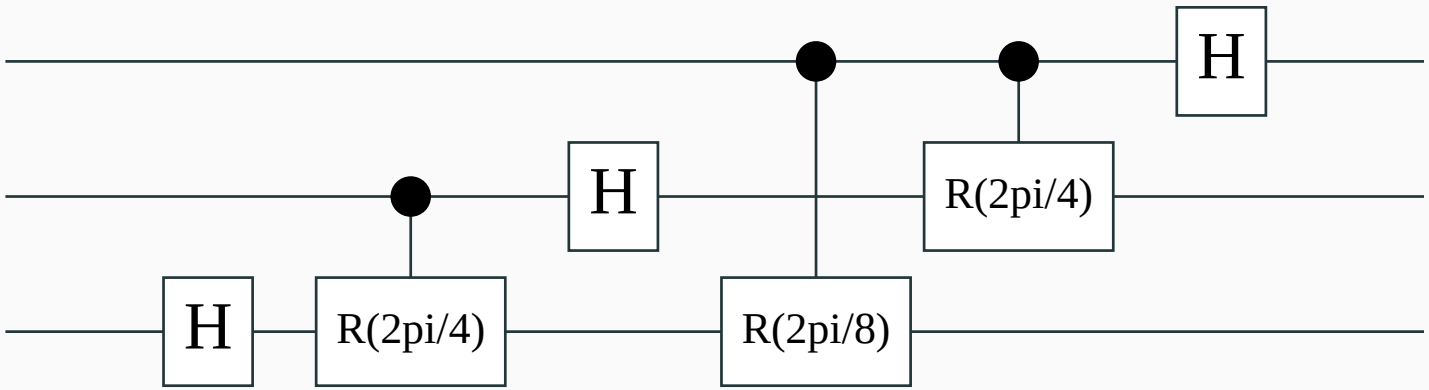


Si el predicado es balanceado la amplitud del estado $|0 \ldots 0\rangle$ se anula, y si es constante se hace ± 1.

Transformada cuántica de Fourier

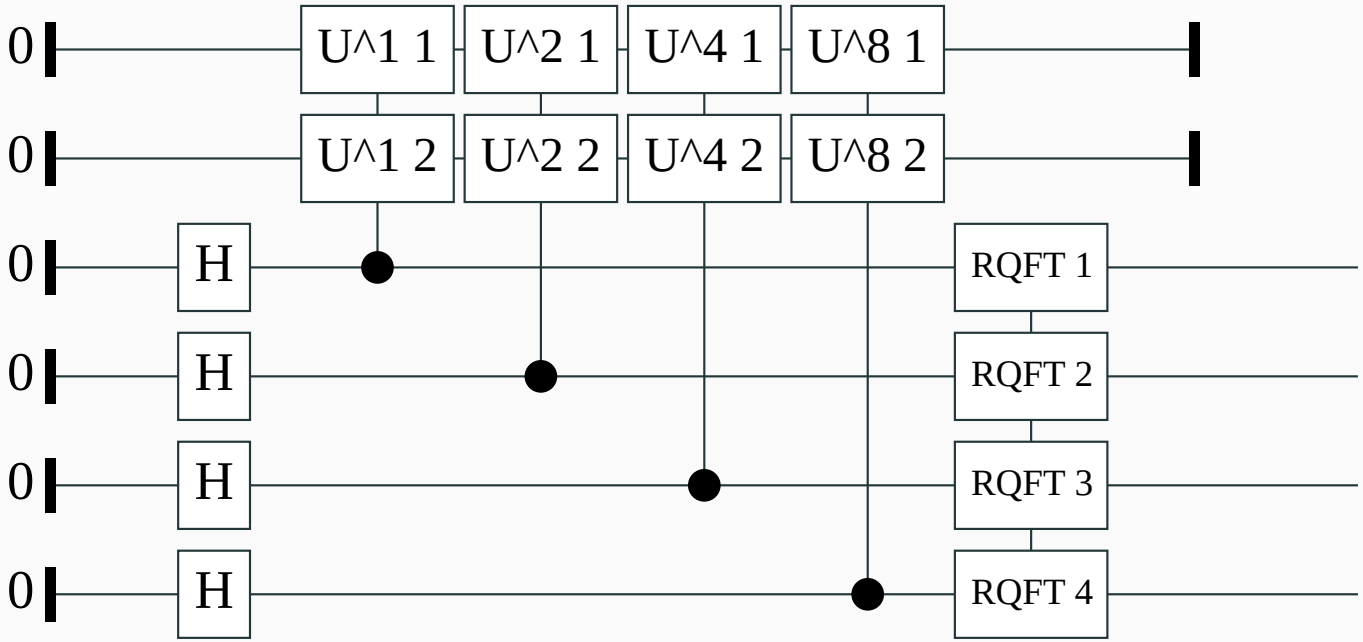
Hace la DFT normalizada sobre las amplitudes en $O((\log N)^2)$ pasos.

```
qft :: [Qubit] → Circ [Qubit]
qft []     = pure []
qft (x:xs) = do
  xs'  ← qft xs
  xs'' ← rotations x xs' (length xs')
  x'   ← hadamard x
  pure (x' : xs'')
```



Algoritmo de estimación de fase

Estima el autovalor asociado a un autovector de un operador unitario en $O((\log N)^2)$ pasos.



Utiliza la transformada de Fourier cuántica inversa.

Algoritmo de Shor

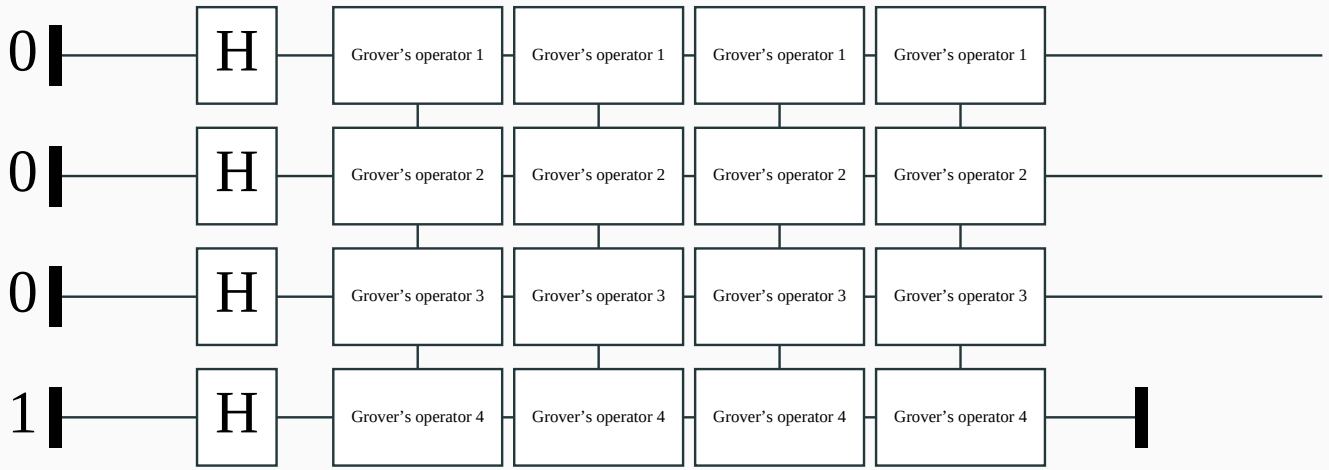
El algoritmo de Shor transforma la factorización de N en el problema de hallar el orden de una unidad $x \in U(\mathbb{Z}_N)$ en tiempo $O((\log N)^3)$.

La simulación no es factible en la práctica; implemento la parte clásica y estimo los recursos para la parte cuántica si hiciera falta.

```
operador :: Integer → Integer → Int → QDInt → Circ QDInt
operador x n j y = do
  q_n           ← qinit (toIntM n)
  (y, z)        ← q_mult_param a y -- x^j mod n * y
  (z, q_n, res) ← q_mod_unsigned z q_n
  pure res -- x^j*y mod n
where a = toIntM (binaryExp x (fromIntegral j) n)
```

Algoritmo de Grover

Encuentra una solución a $f(x) = 1$ en $O\left(\sqrt{N}\right)$ consultas a f.



Se basa en la rotación adecuada de un vector en un plano del espacio de estados. Es factible simularlo en casos pequeños.

Puede apoyarse en el *algoritmo de conteo cuántico*.

Operador de Grover

El oráculo refleja

$$|x\rangle \mapsto (-1)^{f(x)} |x\rangle$$

mientras que el operador `phaseShift` refleja

$$|x\rangle \mapsto (-1)^{\delta_{x0}} |x\rangle .$$

Esto nos da una rotación.

```
diffusion :: [Qubit] → Circ [Qubit]
diffusion = map_hadamard >=> phaseShift >=> map_hadamard

groverOperator ::
   Oracle [Qubit] → ([Qubit], Qubit) → Circ ([Qubit], Qubit)
groverOperator oracle (xs, y) = do
  (xs, y) ← circuit oracle (xs, y)
  xs      ← diffusion xs
  pure (xs, y)
```

Conclusiones

- Los últimos 25 años han supuesto grandes avances en la teoría de la complejidad cuántica, que tiene profundas relaciones con el ámbito clásico; la demostración de la supremacía cuántica resolvería el problema P $\stackrel{?}{=}$ PSPACE.
- Los lenguajes de programación cuánticos permiten expresar los algoritmos presentes en la literatura de forma concisa, pero trabajan aún a muy bajo nivel y no hay consenso respecto de las estructuras de control básicas.

Conclusiones

- Los últimos 25 años han supuesto grandes avances en la teoría de la complejidad cuántica, que tiene profundas relaciones con el ámbito clásico; la demostración de la supremacía cuántica resolvería el problema $\mathrm{P} \stackrel{?}{=} \mathrm{PSPACE}$.
- Los lenguajes de programación cuánticos permiten expresar los algoritmos presentes en la literatura de forma concisa, pero trabajan aún a muy bajo nivel y no hay consenso respecto de las estructuras de control básicas.

Gracias por su atención