量子計算的數學基礎

Mathematical Foundation of Quantum Computing

量子演算法

§1 Deutsch 問題與 Deutsch 演算法

§2 Deutsch-Jozsa 問題與 Deutsch-Jozsa 演算法

§3 搜尋問題與 Grover 演算法

一些背景知識

在進行今天的課程之前，先補充／複習一些背景知識。

1. 我們將一個 n 位元的二進位數寫成 $(j_1 j_2 \cdots j_n)_2$ 或是簡單寫成 $j_1 j_2 \cdots j_n$，也就是說

$$(j_1 j_2 \cdots j_n)_2 = 2^{n-1} j_1 + 2^{n-2} j_2 + \cdots + 2 j_{n-1} + j_n \,.$$

一個 n-qubit 數可寫為 $|j_1 j_2 \cdots j_n\rangle$ 或是 $|j_1 \cdots j_\ell\rangle |j_{\ell+1} \cdots j_n\rangle$（或是可以拆成更多分段）。另外若 $j = (j_1 j_2 \cdots j_n)_2$，我們也將 $|j_1 j_2 \cdots j_n\rangle$ 簡記為 $|j\rangle$。n-qubit 的 $|0\rangle$ 亦被記為 $|0\rangle^{\otimes n}$。

2. 所有 n 位元數所形成的集合被表示成 $\{0,1\}^n$，意即

$$\{0,1\}^n = \left\{ j_1 j_2 \cdots j_n \,\middle|\, j_1, j_2, \cdots, j_n \in \{0,1\} \right\}.$$

3. 所謂的布林函數為從 $\{0,1\}^n$ 映至 $\{0,1\}$ 的函數。
4. 所謂的廣義布林函數從 $\{0,1\}^n$ 映至 $\{0,1\}^m$ 的函數。

一些背景知識

在進行今天的課程之前，先補充／複習一些背景知識。

1. 我們將一個 n 位元的二進位數寫成 $(j_1 j_2 \cdots j_n)_2$ 或是簡單寫成 $j_1 j_2 \cdots j_n$，也就是說

$$(j_1 j_2 \cdots j_n)_2 = 2^{n-1} j_1 + 2^{n-2} j_2 + \cdots + 2 j_{n-1} + j_n \,.$$

一個 n-qubit 數可寫為 $|j_1 j_2 \cdots j_n\rangle$ 或是 $|j_1 \cdots j_\ell\rangle |j_{\ell+1} \cdots j_n\rangle$（或是可以拆成更多分段）。另外若 $j = (j_1 j_2 \cdots j_n)_2$，我們也將 $|j_1 j_2 \cdots j_n\rangle$ 簡記為 $|j\rangle$。n-qubit 的 $|0\rangle$ 亦被記為 $|0\rangle^{\otimes n}$。

2. 所有 n 位元數所形成的集合被表示成 $\{0,1\}^n$，意即

$$\{0,1\}^n = \left\{ j_1 j_2 \cdots j_n \,\middle|\, j_1, j_2, \cdots, j_n \in \{0,1\} \right\} .$$

3. 所謂的布林函數為從 $\{0,1\}^n$ 映至 $\{0,1\}$ 的函數。
4. 所謂的廣義布林函數從 $\{0,1\}^n$ 映至 $\{0,1\}^m$ 的函數。

一些背景知識

在進行今天的課程之前，先補充／複習一些背景知識。

1. 我們將一個 n 位元的二進位數寫成 $(j_1 j_2 \cdots j_n)_2$ 或是簡單寫成 $j_1 j_2 \cdots j_n$，也就是說

$$(j_1 j_2 \cdots j_n)_2 = 2^{n-1} j_1 + 2^{n-2} j_2 + \cdots + 2 j_{n-1} + j_n \,.$$

一個 n-qubit 數可寫為 $|j_1 j_2 \cdots j_n\rangle$ 或是 $|j_1 \cdots j_\ell\rangle |j_{\ell+1} \cdots j_n\rangle$（或是可以拆成更多分段）。另外若 $j = (j_1 j_2 \cdots j_n)_2$，我們也將 $|j_1 j_2 \cdots j_n\rangle$ 簡記為 $|j\rangle$。n-qubit 的 $|0\rangle$ 亦被記為 $|0\rangle^{\otimes n}$。

2. 所有 n 位元數所形成的集合被表示成 $\{0,1\}^n$，意即

$$\{0,1\}^n = \left\{ j_1 j_2 \cdots j_n \,\middle|\, j_1, j_2, \cdots, j_n \in \{0,1\} \right\} .$$

3. 所謂的布林函數為從 $\{0,1\}^n$ 映至 $\{0,1\}$ 的函數。
4. 所謂的廣義布林函數從 $\{0,1\}^n$ 映至 $\{0,1\}^m$ 的函數。

一些背景知識

5. 單一量子態通常寫成 $\alpha_0|0\rangle + \alpha_1|1\rangle$，其中 $|\alpha_j|^2$ 代表測量該量子態出現 j 的機率。因此 $|\alpha_0|^2 + |\alpha_1|^2 = 1$。$\alpha_0$ 和 α_1 被稱作此量子態的機率振幅（probability amplitude）。同理，一個 n-qubit 量子態被表示成

$$|\psi\rangle = \sum_{j=0}^{2^n-1} a_j|j\rangle = \sum_{j_1, j_2, \cdots, j_n \in \{0,1\}} a_{j_1 j_2 \cdots j_n} |j_1 j_2 \cdots j_n\rangle$$
$$= a_{0\cdots0}|0\cdots0\rangle + a_{0\cdots01}|0\cdots01\rangle + \cdots + a_{1\cdots1}|1\cdots1\rangle,$$

在此 $|a_{j_1 j_2 \cdots j_n}|^2$ 代表測量該量子態出現 $|j_1 j_2 \cdots j_n\rangle$ 的機率，因此 $\sum_{j_1, j_2, \cdots, j_n \in \{0,1\}} |a_{j_1 j_2 \cdots j_n}|^2 = 1$。這些係數被稱作此量子態的機率振幅。因為一般而言這些係數都是複數，然後一個 n-qubit 的量子態有 2^n 個係數，所以我們以 $\mathbb{C}^{2^n}$ 來表示所有 n-qubit 量子態（所形成的集合）。

一些背景知識

5. 單一量子態通常寫成 $\alpha_0|0\rangle+\alpha_1|1\rangle$，其中 $|\alpha_j|^2$ 代表測量該量子態出現 j 的機率。因此 $|\alpha_0|^2+|\alpha_1|^2=1$。$\alpha_0$ 和 α_1 被稱作此量子態的機率振幅（probability amplitude）。同理，一個 n-qubit 量子態被表示成

$$|\psi\rangle=\sum_{j=0}^{2^n-1}a_j|j\rangle=\sum_{j_1,j_2,\cdots,j_n\in\{0,1\}}a_{j_1j_2\cdots j_n}|j_1j_2\cdots j_n\rangle$$
$$=a_{0\cdots0}|0\cdots0\rangle+a_{0\cdots01}|0\cdots01\rangle+\cdots+a_{1\cdots1}|1\cdots1\rangle,$$

在此 $|a_{j_1j_2\cdots j_n}|^2$ 代表測量該量子態出現 $|j_1j_2\cdots j_n\rangle$ 的機率，因此 $\sum_{j_1,j_2,\cdots,j_n\in\{0,1\}}|a_{j_1j_2\cdots j_n}|^2=1$。這些係數被稱作此量子態的機率振幅。因為一般而言這些係數都是複數，然後一個 n-qubit 的量子態有 2^n 個係數，所以我們以 $\mathbb{C}^{2^n}$ 來表示所有 n-qubit 量子態（所形成的集合）。

一些背景知識

5. 單一量子態通常寫成 $\alpha_0|0\rangle + \alpha_1|1\rangle$，其中 $|\alpha_j|^2$ 代表測量該量子態出現 j 的機率。因此 $|\alpha_0|^2 + |\alpha_1|^2 = 1$。$\alpha_0$ 和 α_1 被稱作此量子態的機率振幅（probability amplitude）。同理，一個 n-qubit 量子態被表示成

$$|\psi\rangle = \sum_{j=0}^{2^n-1} a_j|j\rangle = \sum_{j_1,j_2,\cdots,j_n\in\{0,1\}} a_{j_1j_2\cdots j_n}|j_1j_2\cdots j_n\rangle$$
$$= a_{0\cdots0}|0\cdots0\rangle + a_{0\cdots01}|0\cdots01\rangle + \cdots + a_{1\cdots1}|1\cdots1\rangle,$$

在此 $|a_{j_1j_2\cdots j_n}|^2$ 代表測量該量子態出現 $|j_1j_2\cdots j_n\rangle$ 的機率，因此 $\sum_{j_1,j_2,\cdots,j_n\in\{0,1\}} |a_{j_1j_2\cdots j_n}|^2 = 1$。這些係數被稱作此量子態的機率振幅。因為一般而言這些係數都是複數，然後一個 n-qubit 的量子態有 2^n 個係數，所以我們以 $\mathbb{C}^{2^n}$ 來表示所有 n-qubit 量子態（所形成的集合）。

一些背景知識

5. 兩個單一量子態的 tensor product 的計算：

$$
\begin{aligned}
&(\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\beta_0|0\rangle + \beta_1|1\rangle) \\
&\quad = \alpha_0\beta_0|0\rangle \otimes |0\rangle + \alpha_0\beta_1|0\rangle \otimes |1\rangle + \alpha_1\beta_0|1\rangle \otimes |0\rangle \\
&\qquad + \alpha_1\beta_1|1\rangle \otimes |1\rangle \\
&\quad \equiv \alpha_0\beta_0|0\rangle|0\rangle + \alpha_0\beta_1|0\rangle|1\rangle + \alpha_1\beta_0|1\rangle|0\rangle + \alpha_1\beta_1|1\rangle|1\rangle .
\end{aligned}
$$

上述的計算類似於展開 $(\alpha_0 x_0 + \alpha_1 x_1)(\beta_0 y_0 + \beta_1 y_1)$ 兩個多項式的乘積，唯獨在此記得 x_0, x_1 要寫在 y_0, y_1 的前面。多個 n-qubit 量子態的 tensor product 也是用分配律來做類似的計算。

一些背景知識

7. Hadamard 閘 H 可以作用在單一量子上面，其效果為

$$\mathrm{H}|j\rangle = \frac{1}{\sqrt{2}} \sum_{k=0}^{1} (-1)^{jk} |k\rangle \qquad \forall\, j \in \{0,1\}$$

且由於線性的性質：

$$\mathrm{H}(\alpha_0|0\rangle + \alpha_1|1\rangle) = \alpha_0 \mathrm{H}|0\rangle + \alpha_1 \mathrm{H}|1\rangle \qquad \forall\, \alpha_0, \alpha_1 \in \mathbb{C}\,.$$

8. CNOT 閘可以作用在兩個量子上面，其效果為

$$\mathbf{CNOT}|x\rangle|y\rangle = |x\rangle|x \oplus y\rangle, \qquad \forall\, x, y \in \{0,1\}$$

且由於線性的性質：對任意的 $\alpha_0, \alpha_1, \beta_0, \beta_1 \in \mathbb{C}$，

$$\begin{aligned}
&\mathbf{CNOT}\big[(\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\beta_0|0\rangle + \beta_1|1\rangle)\big] \\
&\quad = \mathbf{CNOT}\big(\alpha_0\beta_0|0\rangle|0\rangle + \alpha_0\beta_1|0\rangle|1\rangle + \alpha_1\beta_0|1\rangle|0\rangle + \alpha_1\beta_1|1\rangle|1\rangle\big) \\
&\quad = \alpha_0\beta_0|0\rangle|0\rangle + \alpha_0\beta_1|0\rangle|1\rangle + \alpha_1\beta_0|1\rangle|1\rangle + \alpha_1\beta_1|1\rangle|0\rangle\,.
\end{aligned}$$

一些背景知識

7. Hadamard 閘 H 可以作用在單一量子上面，其效果為

$$\mathrm{H}|j\rangle = \frac{1}{\sqrt{2}} \sum_{k=0}^{1} (-1)^{jk} |k\rangle \qquad \forall\, j \in \{0,1\}$$

且由於線性的性質：

$$\mathrm{H}(\alpha_0|0\rangle + \alpha_1|1\rangle) = \alpha_0 \mathrm{H}|0\rangle + \alpha_1 \mathrm{H}|1\rangle \qquad \forall\, \alpha_0, \alpha_1 \in \mathbb{C}\,.$$

8. **CNOT** 閘可以作用在兩個量子上面，其效果為

$$\mathbf{CNOT}|x\rangle|y\rangle = |x\rangle|x \oplus y\rangle, \qquad \forall\, x, y \in \{0,1\}$$

且由於線性的性質：對任意的 $\alpha_0, \alpha_1, \beta_0, \beta_1 \in \mathbb{C}$，

$$\begin{aligned}
&\mathbf{CNOT}\big[(\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\beta_0|0\rangle + \beta_1|1\rangle)\big] \\
&= \mathbf{CNOT}\big(\alpha_0\beta_0|0\rangle|0\rangle + \alpha_0\beta_1|0\rangle|1\rangle + \alpha_1\beta_0|1\rangle|0\rangle + \alpha_1\beta_1|1\rangle|1\rangle\big) \\
&= \alpha_0\beta_0|0\rangle|0\rangle + \alpha_0\beta_1|0\rangle|1\rangle + \alpha_1\beta_0|1\rangle|1\rangle + \alpha_1\beta_1|1\rangle|0\rangle\,.
\end{aligned}$$

一些背景知識

9. 量子電路是包含了輸入節點、內部節點和輸出節點的有限、有向（從左至右）不循環圖。
   (1) 每條橫線代表一個量子位元（隨時間演化的狀態）。
   (2) 在量子電路的最左邊有數個初始輸入節點（此外還可能還有更多的輸入節點 $|0\rangle$ 當作 workspace）。
   (3) 量子電路的內部節點是量子邏輯閘，每個量子邏輯閘最多在作用在 2 個量子比特上。這些邏輯閘將初始量子態轉換為最終量子態，而最終量子態通常是一個疊加態。
   (4) 當一個量子位元（控制位元）的值決定了是否在另一個量子位元（目標位元）運作某邏輯閘，則在控制位元節點上畫上一點、在目標位元節點上畫上對應的邏輯閘，且將點與該邏輯閘以垂直線連接。
   (5) 我們測量這個最終量子態的一些輸出節點以獲得答案。

一些背景知識

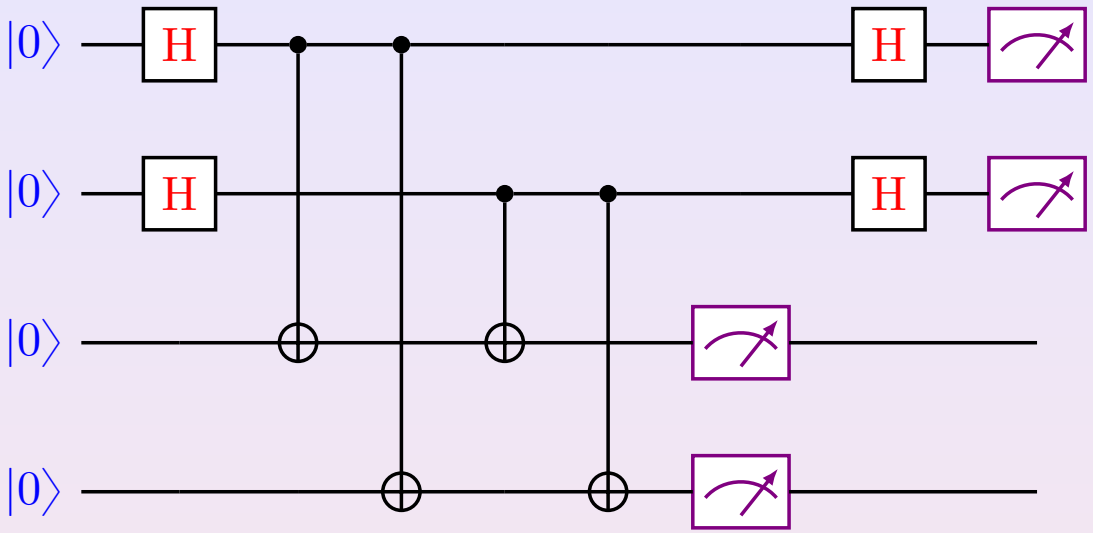


Figure 1: 某個使用 Simon 演算法的量子電路

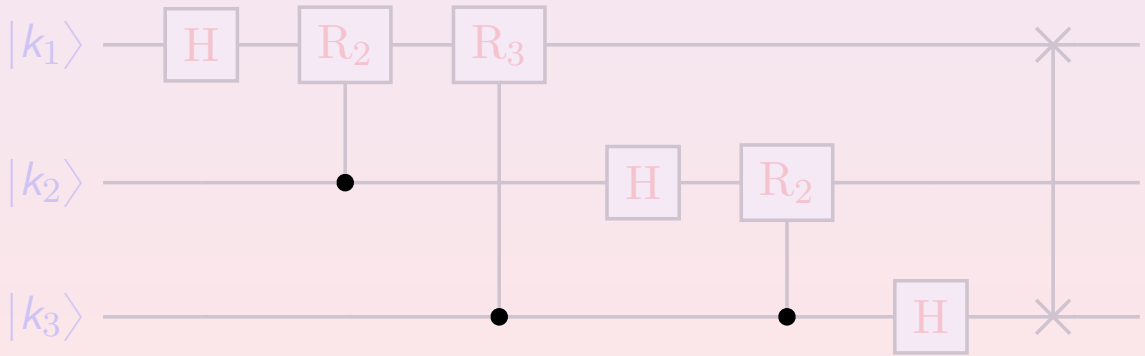


Figure 2: 3-qubit 的量子 Fourier 轉換（QFT）之量子電路

一些背景知識

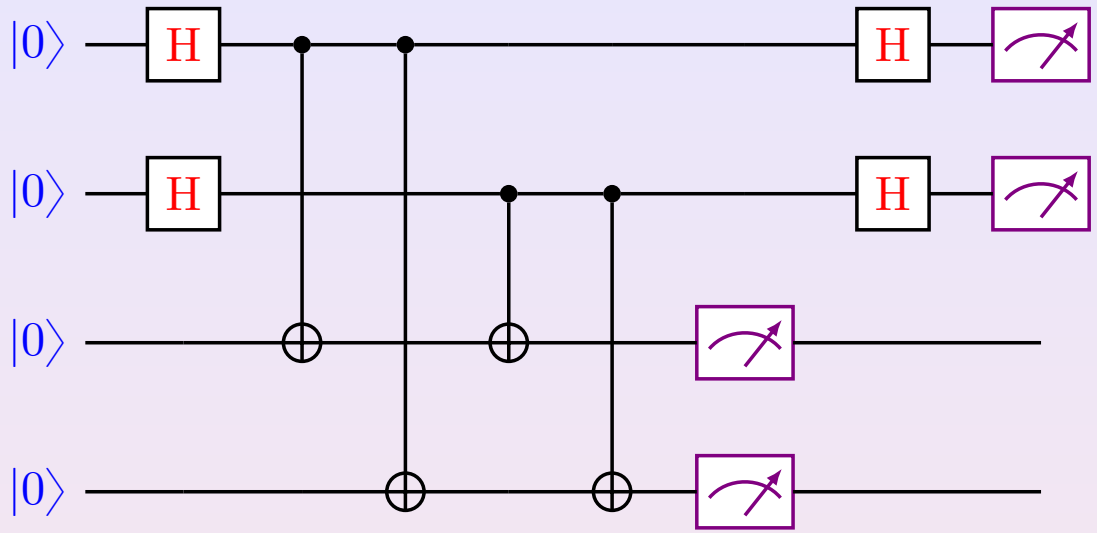


Figure 1: 某個使用 Simon 演算法的量子電路

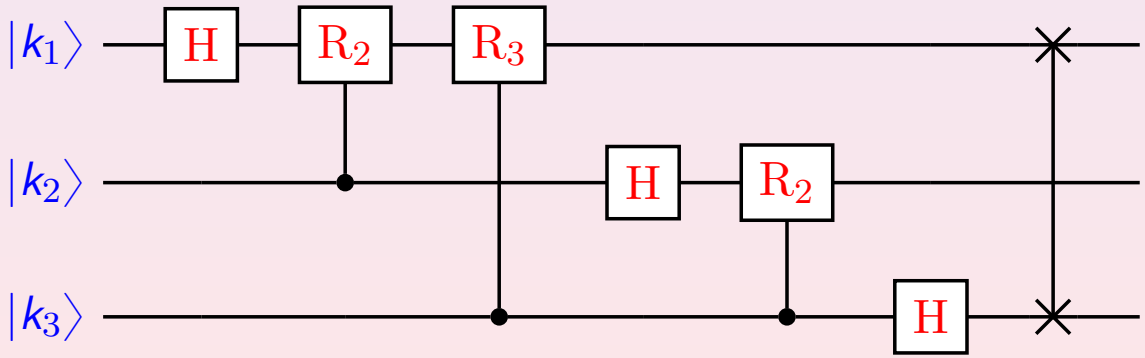


Figure 2: 3-qubit 的量子 Fourier 轉換（QFT）之量子電路

一些背景知識

當有模組是已知或可用時，也可以直接把模組用到量子電路中：

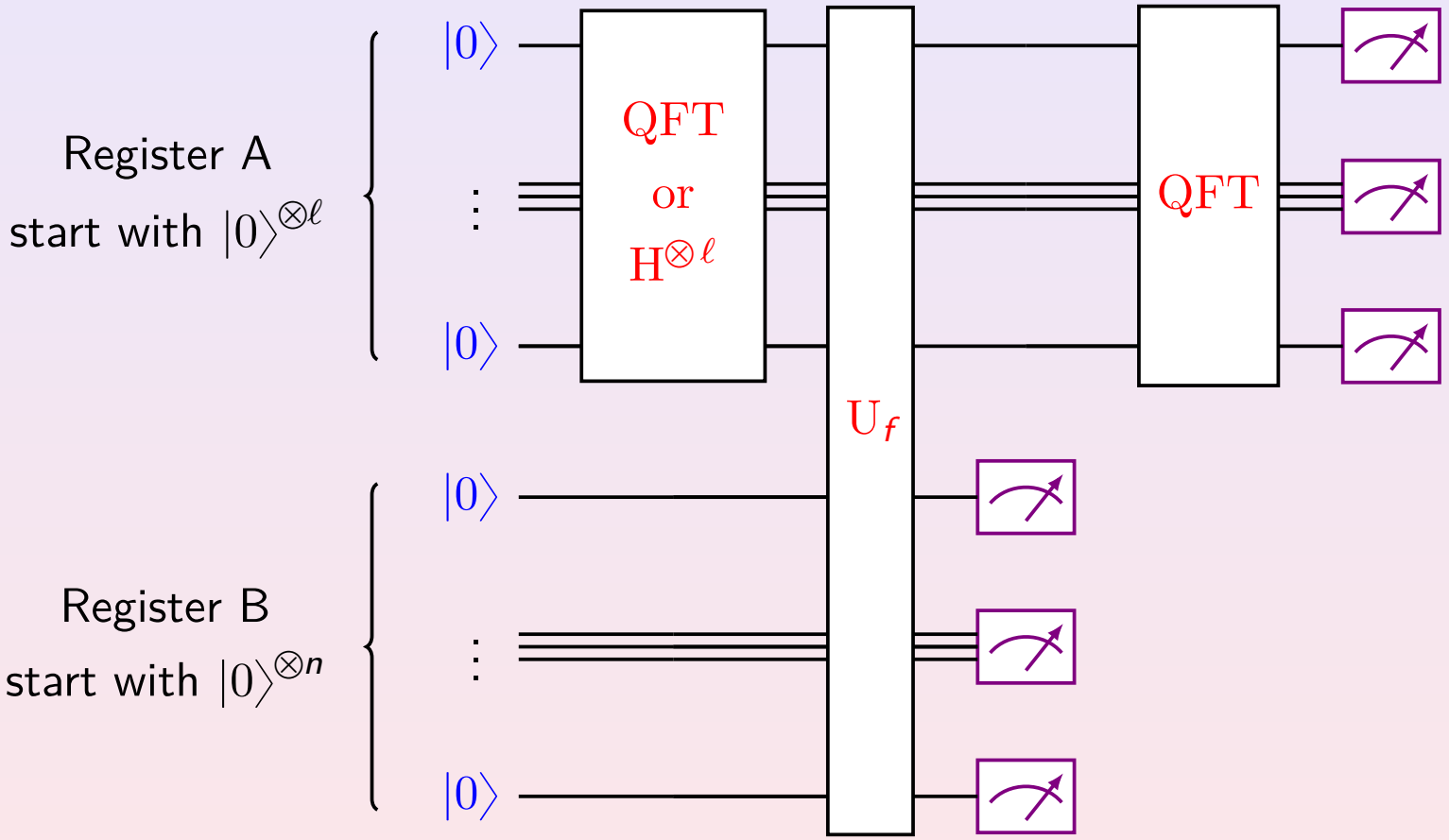


Figure 3: Shor 演算法的量子電路

一些背景知識

10. 在上述 Shor 演算法的量子電路中出現了 $\mathrm{H}^{\otimes \ell}$ 這個操作，其意義是對 ℓ 個量子同時操作 Hadamard 閘 H。例如

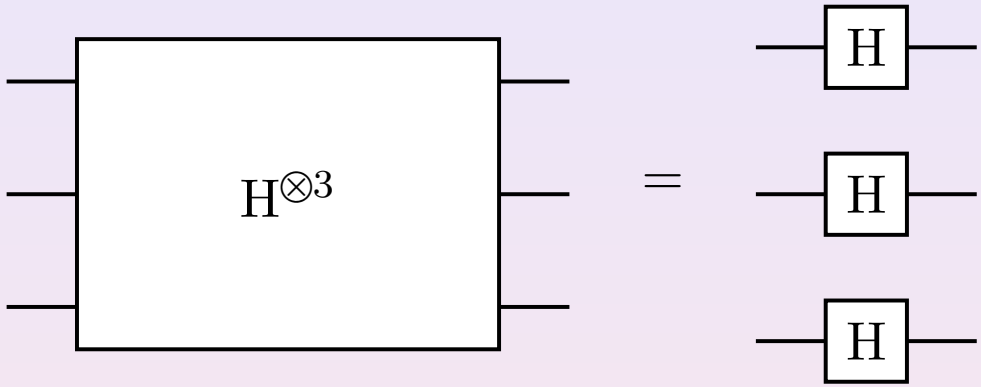


Figure 4: $\mathrm{H}^{\otimes 3}$ 等於三個量子同時作用 Hadamard 閘

11. 一般而言，在同一時間點對 n 個不同量子位元進行 A_1, A_2, $\cdots$, A_n 的操作，則在此時間點所進行的操作被記為 $\mathrm{A}_1 \otimes \mathrm{A}_2 \otimes \cdots \otimes \mathrm{A}_n$，也就是說

$$(\mathrm{A}_1 \otimes \mathrm{A}_2 \otimes \cdots \otimes \mathrm{A}_n)(|x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle)$$
$$= (\mathrm{A}_1|x_1\rangle) \otimes (\mathrm{A}_2|x_2\rangle) \otimes \cdots \otimes (\mathrm{A}_n|x_n\rangle) .$$

一些背景知識

10. 在上述 Shor 演算法的量子電路中出現了 $\mathrm{H}^{\otimes \ell}$ 這個操作，其意義是對 ℓ 個量子同時操作 Hadamard 閘 H。例如

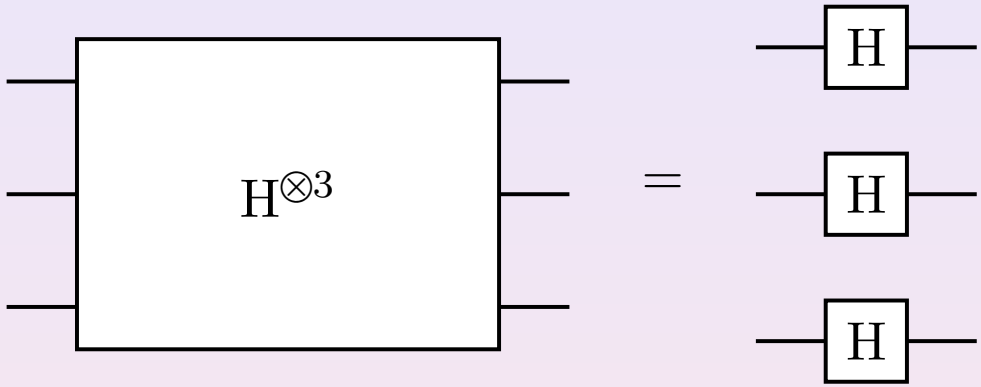


Figure 4: $\mathrm{H}^{\otimes 3}$ 等於三個量子同時作用 Hadamard 閘

11. 一般而言，在同一時間點對 n 個不同量子位元進行 A_1, A_2, $\cdots$, A_n 的操作，則在此時間點所進行的操作被記為 $\mathrm{A}_1 \otimes \mathrm{A}_2 \otimes \cdots \otimes \mathrm{A}_n$，也就是說

$$(\mathrm{A}_1 \otimes \mathrm{A}_2 \otimes \cdots \otimes \mathrm{A}_n)(|x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle)$$
$$= (\mathrm{A}_1|x_1\rangle) \otimes (\mathrm{A}_2|x_2\rangle) \otimes \cdots \otimes (\mathrm{A}_n|x_n\rangle)\,.$$

一些背景知識

很多量子演算法都會與一個所謂問詢 (query) 的機制有關。給定一個（廣義）布林函數 $f\colon \{0,1\}^n \to \{0,1\}^m$，所謂的 query 指的是函數 f 的取值（一個 query 就是指一次函數取值）。更具體的說，我們常常考慮以下的映射

$$U_f\colon |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle \qquad \forall\, x \in \{0,1\}^n, y \in \{0,1\}^m,$$

在此 $\oplus$ 表示 bitwise exclusive-or (**XOR**) 或是 $\mathbb{Z}_2$ 中的加法 (addition modulo 2)，例如 $01 \oplus 11 = 10$。而很多量子演算法都與這個 U_f 有關。

Example

1. Deutsch、Deutsch-Jozsa 與 Grover 演算法是 $m = 1$ 的 case。
2. Simon、Shor 演算法是 $m > 1$ 的 case。

一些背景知識

很多量子演算法都會與一個所謂問詢 (query) 的機制有關。給定一個（廣義）布林函數 $f\colon \{0,1\}^n \to \{0,1\}^m$，所謂的 query 指的是函數 f 的取值（一個 query 就是指一次函數取值）。更具體的說，我們常常考慮以下的映射

$$\mathrm{U}_f\colon |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle \qquad \forall\, x \in \{0,1\}^n, y \in \{0,1\}^m,$$

在此 $\oplus$ 表示 bitwise exclusive-or (**XOR**) 或是 $\mathbb{Z}_2$ 中的加法 (addition modulo 2)，例如 $01 \oplus 11 = 10$。而很多量子演算法都與這個 U_f 有關。

Example

1. Deutsch、Deutsch-Jozsa 與 Grover 演算法是 $m = 1$ 的 case。
2. Simon、Shor 演算法是 $m > 1$ 的 case。

一些背景知識

很多量子演算法都會與一個所謂問詢 (query) 的機制有關。給定一個（廣義）布林函數 $f\colon \{0,1\}^n \to \{0,1\}^m$，所謂的 query 指的是函數 f 的取值（一個 query 就是指一次函數取值）。更具體的說，我們常常考慮以下的映射

$$U_f\colon |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle \qquad \forall\, x \in \{0,1\}^n, y \in \{0,1\}^m,$$

在此 $\oplus$ 表示 bitwise exclusive-or (**XOR**) 或是 $\mathbb{Z}_2$ 中的加法 (addition modulo 2)，例如 $01 \oplus 11 = 10$。而很多量子演算法都與這個 U_f 有關。

Example

1. Deutsch、Deutsch-Jozsa 與 Grover 演算法是 $m = 1$ 的 case。
2. Simon、Shor 演算法是 $m > 1$ 的 case。

一些背景知識

1. 我們注意到若 $f\colon \{0,1\}^n \to \{0,1\}^m$ 是一個（廣義）布林函數，則 $U_f\colon \{0,1\}^{n+m} \to \{0,1\}^{n+m}$ 也是一個廣義布林函數。
2. 在古典邏輯電路理論中，任何一個（廣義）布林函數都可以用 AND、OR 和 NOT 三個基本邏輯閘（加上填補與限制／重排功能）來實現。然而如何在量子電路上實現 U_f 目前沒有一般法則可以依循，因此 U_f 被視為一個上帝所給的「黑盒子」，意即不知運作原理但是卻能拿來使用的工具。
3. 在現行的量子演算法中，我們都假設「給定任一（廣義）布林函數 $f\colon \{0,1\}^n \to \{0,1\}^m$ 其對應的線性化 U_f 可被實現」。
4. 在量子電路中一個 U_f 對應著一次的 query。許多量子演算法相較於古典演算法的優勢即在於這些量子演算法所需的 query 次數遠小於古典演算法。

一些背景知識

1. 我們注意到若 $f\colon \{0,1\}^n \to \{0,1\}^m$ 是一個（廣義）布林函數，則 $U_f\colon \{0,1\}^{n+m} \to \{0,1\}^{n+m}$ 也是一個廣義布林函數。
2. 在古典邏輯電路理論中，任何一個（廣義）布林函數都可以用 **AND**、**OR** 和 **NOT** 三個基本邏輯閘（加上填補與限制／重排功能）來實現。然而如何在量子電路上實現 U_f 目前沒有一般法則可以依循，因此 U_f 被視為一個上帝所給的「黑盒子」，意即不知運作原理但是卻能拿來使用的工具。
3. 在現行的量子演算法中，我們都假設「給定任一（廣義）布林函數 $f\colon \{0,1\}^n \to \{0,1\}^m$ 其對應的線性化 U_f 可被實現」。
4. 在量子電路中一個 U_f 對應著一次的 query。許多量子演算法相較於古典演算法的優勢即在於這些量子演算法所需的 query 次數遠小於古典演算法。

一些背景知識

1. 我們注意到若 $f\colon \{0,1\}^n \to \{0,1\}^m$ 是一個（廣義）布林函數，則 $U_f\colon \{0,1\}^{n+m} \to \{0,1\}^{n+m}$ 也是一個廣義布林函數。
2. 在古典邏輯電路理論中，任何一個（廣義）布林函數都可以用 **AND**、**OR** 和 **NOT** 三個基本邏輯閘（加上填補與限制／重排功能）來實現。然而如何在量子電路上實現 U_f 目前沒有一般法則可以依循，因此 U_f 被視為一個上帝所給的「黑盒子」，意即不知運作原理但是卻能拿來使用的工具。
3. 在現行的量子演算法中，我們都假設「給定任一（廣義）布林函數 $f\colon \{0,1\}^n \to \{0,1\}^m$ 其對應的線性化 U_f 可被實現」。
4. 在量子電路中一個 U_f 對應著一次的 query。許多量子演算法相較於古典演算法的優勢即在於這些量子演算法所需的 query 次數遠小於古典演算法。

一些背景知識

1. 我們注意到若 $f\colon \{0,1\}^n \to \{0,1\}^m$ 是一個（廣義）布林函數，則 $\mathrm{U}_f\colon \{0,1\}^{n+m} \to \{0,1\}^{n+m}$ 也是一個廣義布林函數。
2. 在古典邏輯電路理論中，任何一個（廣義）布林函數都可以用 **AND**、**OR** 和 **NOT** 三個基本邏輯閘（加上填補與限制／重排功能）來實現。然而如何在量子電路上實現 U_f 目前沒有一般法則可以依循，因此 U_f 被視為一個上帝所給的「黑盒子」，意即不知運作原理但是卻能拿來使用的工具。
3. 在現行的量子演算法中，我們都假設「給定任一（廣義）布林函數 $f\colon \{0,1\}^n \to \{0,1\}^m$ 其對應的線性化 U_f 可被實現」。
4. 在量子電路中一個 U_f 對應著一次的 query。許多量子演算法相較於古典演算法的優勢即在於這些量子演算法所需的 query 次數遠小於古典演算法。

一些背景知識

1. 我們注意到若 $f\colon \{0,1\}^n \to \{0,1\}^m$ 是一個（廣義）布林函數，則 $U_f\colon \{0,1\}^{n+m} \to \{0,1\}^{n+m}$ 也是一個廣義布林函數。
2. 在古典邏輯電路理論中，任何一個（廣義）布林函數都可以用 **AND**、**OR** 和 **NOT** 三個基本邏輯閘（加上填補與限制／重排功能）來實現。然而如何在量子電路上實現 U_f 目前沒有一般法則可以依循，因此 U_f 被視為一個上帝所給的「黑盒子」，意即不知運作原理但是卻能拿來使用的工具。
3. 在現行的量子演算法中，我們都假設「給定任一（廣義）布林函數 $f\colon \{0,1\}^n \to \{0,1\}^m$ 其對應的線性化 U_f 可被實現」。
4. 在量子電路中一個 U_f 對應著一次的 query。許多量子演算法相較於古典演算法的優勢即在於這些量子演算法所需的 query 次數遠小於古典演算法。

一些背景知識

Example

針對特別的布林函數 f 有時還是有機會找出對應的量子電路。例如考慮 $f(x_1x_2\cdots x_n)=x_n$ 這樣由 $\{0,1\}^n$ 映至 $\{0,1\}$ 的布林函數（這個布林函數是吐出輸入二進位數的最低位元）。若 $\mathbf{I}_{n-1}$ 代表 $(n-1)$-qubit 系統中的 identity map，那麼

$$\begin{aligned}
&(\mathbf{I}_{n-1}\otimes \mathbf{CNOT})(|x\rangle|y\rangle)\\
&\quad=(\mathbf{I}_{n-1}\otimes \mathbf{CNOT})(|x_1\cdots x_{n-1}x_n\rangle|y\rangle)\\
&\quad=(\mathbf{I}_{n-1}\otimes \mathbf{CNOT})(|x_1\cdots x_{n-1}\rangle|x_ny\rangle)\\
&\quad=\left(\mathbf{I}_{n-1}|x_1\cdots x_{n-1}\rangle\right)\otimes\left(\mathbf{CNOT}(|x_n\rangle|y\rangle)\right)\\
&\quad=|x_1\cdots x_{n-1}\rangle|x_n\rangle|y\oplus x_n\rangle=|x_1\cdots x_n\rangle|y\oplus x_n\rangle\\
&\quad=|x\rangle|y\oplus f(x)\rangle.
\end{aligned}$$

也就是說，$\mathbf{U}_f=\mathbf{I}_{n-1}\otimes\mathbf{CNOT}$.

一些背景知識

Example

針對特別的布林函數 f 有時還是有機會找出對應的量子電路。例如考慮 $f(x_1x_2\cdots x_n)=x_n$ 這樣由 $\{0,1\}^n$ 映至 $\{0,1\}$ 的布林函數（這個布林函數是吐出輸入二進位數的最低位元）。若 $\mathbf{I}_{n-1}$ 代表 $(n-1)$-qubit 系統中的 identity map，那麼

$$\begin{aligned}
&(\mathbf{I}_{n-1}\otimes \mathbf{CNOT})(|x\rangle|y\rangle)\\
&\quad=(\mathbf{I}_{n-1}\otimes \mathbf{CNOT})(|x_1\cdots x_{n-1}x_n\rangle|y\rangle)\\
&\quad=(\mathbf{I}_{n-1}\otimes \mathbf{CNOT})(|x_1\cdots x_{n-1}\rangle|x_n y\rangle)\\
&\quad=\left(\mathbf{I}_{n-1}|x_1\cdots x_{n-1}\rangle\right)\otimes\left(\mathbf{CNOT}(|x_n\rangle|y\rangle)\right)\\
&\quad=|x_1\cdots x_{n-1}\rangle|x_n\rangle|y\oplus x_n\rangle=|x_1\cdots x_n\rangle|y\oplus x_n\rangle\\
&\quad=|x\rangle|y\oplus f(x)\rangle\,.
\end{aligned}$$

也就是說，$\mathrm{U}_f=\mathbf{I}_{n-1}\otimes\mathbf{CNOT}$.

一些背景知識

Example (con't)

因此，U_f 可由以下的量子電路實現

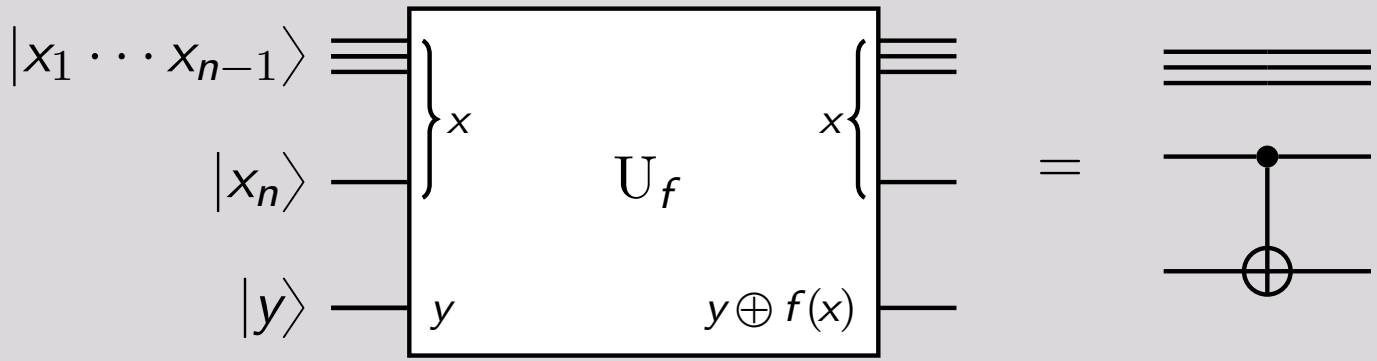
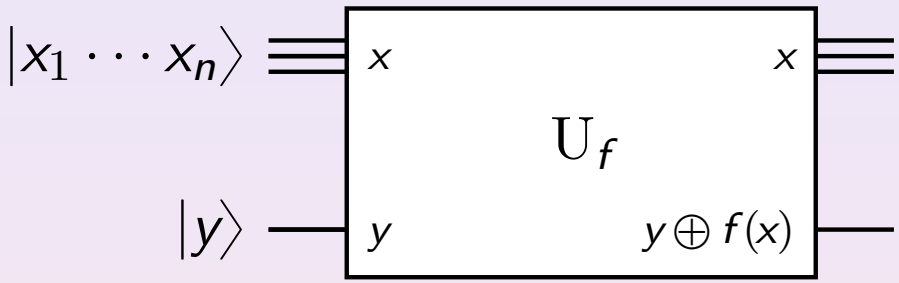


Figure 5: 對於布林函數 $f(x_1, \cdots, x_n) = x_n$ 所對應之 U_f 的量子電路

一些背景知識

下課十分鐘習題：

給定一布林函數 $f: \{0,1\}^n \to \{0,1\}$ 以及其對應的 U_f 之量子電路



請問建立 U_{1-f} 的量子電路。

一些背景知識

假設對於給定的布林函數 $f\colon \{0,1\}^n \to \{0,1\}$ 我們有辦法實現 U_f 的量子電路，那麼我們就有辦法據此製造出 $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ 這樣的機制。此機制是將 $|-\rangle \equiv \mathrm{H}|1\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big)$ 代入 U_f 所需輸入的變數的最後一個 qubit state 而得：

$$\mathrm{U}_f\big(|x\rangle|-\rangle\big) = |x\rangle\frac{1}{\sqrt{2}}\big(|f(x)\rangle - |1\oplus f(x)\rangle\big) = (-1)^{f(x)}|x\rangle|-\rangle.$$

上述 $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ 是改變量子態相位的機制：當 $f(x)$ 為 1 時則在量子態 $|x\rangle$ 的機率振幅多乘了 -1；而當 $f(x)$ 為 0 時則維持原機率振幅。這個所謂的 “phase-oracle” 在量子計算中比起傳統的 query 更為方便。

在以下的討論，我們以 $\mathrm{U}_{f,\pm}$ 來表示 $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ 這個映射。

一些背景知識

假設對於給定的布林函數 $f\colon \{0,1\}^n \to \{0,1\}$ 我們有辦法實現 U_f 的量子電路，那麼我們就有辦法據此製造出 $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ 這樣的機制。此機制是將 $|-\rangle \equiv \mathrm{H}|1\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big)$ 代入 U_f 所需輸入的變數的最後一個 qubit state 而得：

$$\mathrm{U}_f\big(|x\rangle|-\rangle\big) = |x\rangle\frac{1}{\sqrt{2}}\big(|f(x)\rangle - |1\oplus f(x)\rangle\big) = (-1)^{f(x)}|x\rangle|-\rangle.$$

上述 $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ 是改變量子態相位的機制：當 $f(x)$ 為 1 時則在量子態 $|x\rangle$ 的機率振幅多乘了 -1；而當 $f(x)$ 為 0 時則維持原機率振幅。這個所謂的 “phase-oracle” 在量子計算中比起傳統的 query 更為方便。

在以下的討論，我們以 $\mathrm{U}_{f,\pm}$ 來表示 $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ 這個映射。

一些背景知識

假設對於給定的布林函數 $f\colon \{0,1\}^n \to \{0,1\}$ 我們有辦法實現 U_f 的量子電路，那麼我們就有辦法據此製造出 $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ 這樣的機制。此機制是將 $|-\rangle \equiv \mathrm{H}|1\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big)$ 代入 U_f 所需輸入的變數的最後一個 qubit state 而得：

$$\mathrm{U}_f\big(|x\rangle|-\rangle\big) = |x\rangle\frac{1}{\sqrt{2}}\big(|f(x)\rangle - |1\oplus f(x)\rangle\big) = (-1)^{f(x)}|x\rangle|-\rangle.$$

上述 $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ 是改變量子態相位的機制：當 $f(x)$ 為 1 時則在量子態 $|x\rangle$ 的機率振幅多乘了 -1；而當 $f(x)$ 為 0 時則維持原機率振幅。這個所謂的 “phase-oracle” 在量子計算中比起傳統的 query 更為方便。

在以下的討論，我們以 $\mathrm{U}_{f,\pm}$ 來表示 $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ 這個映射。

一些背景知識

假設對於給定的布林函數 $f\colon \{0,1\}^n \to \{0,1\}$ 我們有辦法實現 U_f 的量子電路，那麼我們就有辦法據此製造出 $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ 這樣的機制。此機制是將 $|-\rangle \equiv \mathrm{H}|1\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big)$ 代入 U_f 所需輸入的變數的最後一個 qubit state 而得：

$$\mathrm{U}_f\big(|x\rangle|-\rangle\big) = |x\rangle\frac{1}{\sqrt{2}}\big(|f(x)\rangle - |1\oplus f(x)\rangle\big) = (-1)^{f(x)}|x\rangle|-\rangle.$$

上述 $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ 是改變量子態相位的機制：當 $f(x)$ 為 1 時則在量子態 $|x\rangle$ 的機率振幅多乘了 -1；而當 $f(x)$ 為 0 時則維持原機率振幅。這個所謂的 “phase-oracle” 在量子計算中比起傳統的 query 更為方便。

在以下的討論，我們以 $\mathrm{U}_{f,\pm}$ 來表示 $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ 這個映射。

§1.1 Deutsch 問題

Deutsch 問題: 給定一布林函數 $f\colon \{0,1\} \to \{0,1\}$。

1. 若 $f(0) = f(1)$，此類的 f 被稱為 "**constant**" function；
2. 若 $f(0) \neq f(1)$，此類的 f 被稱為 "**balanced**" function。

Deutsch 問題的目標是找出 f 是 constant 或是 balanced。

傳統演算法的 query 次數估計：

1. 任何古典演算法都不可能在只取值一次之後就知道 f 為 constant 或是 balanced，一定得要進行兩次 queries（即函數取值）之後才能確定知道 f 是哪一類的函數。
2. 有四個不同的函數 $f\colon \{0,1\} \to \{0,1\}$，其中有兩個是 constant function 有兩個是 balanced function，在不做任何取值的情況下盲猜的結果有 $1/2$ 的機會能猜中 f 是哪一類函數。

§1.1 Deutsch 問題

Deutsch 問題: 給定一布林函數 $f\colon \{0,1\} \to \{0,1\}$。

1. 若 $f(0) = f(1)$，此類的 f 被稱為 "**constant**" function；
2. 若 $f(0) \neq f(1)$，此類的 f 被稱為 "**balanced**" function。

Deutsch 問題的目標是找出 f 是 constant 或是 balanced。

傳統演算法的 query 次數估計：

1. 任何古典演算法都不可能在只取值一次之後就知道 f 為 constant 或是 balanced，一定得要進行兩次 queries（即函數取值）之後才能確定知道 f 是哪一類的函數。
2. 有四個不同的函數 $f\colon \{0,1\} \to \{0,1\}$，其中有兩個是 constant function 有兩個是 balanced function，在不做任何取值的情況下盲猜的結果有 $1/2$ 的機會能猜中 f 是哪一類函數。

§1.1 Deutsch 問題

Deutsch 問題: 給定一布林函數 $f\colon \{0,1\} \to \{0,1\}$。

1. 若 $f(0) = f(1)$，此類的 f 被稱為 "**constant**" function；
2. 若 $f(0) \neq f(1)$，此類的 f 被稱為 "**balanced**" function。

Deutsch 問題的目標是找出 f 是 constant 或是 balanced。

傳統演算法的 query 次數估計：

1. 任何古典演算法都不可能在只取值一次之後就知道 f 為 constant 或是 balanced，一定得要進行兩次 queries（即函數取值）之後才能確定知道 f 是哪一類的函數。
2. 有四個不同的函數 $f\colon \{0,1\} \to \{0,1\}$，其中有兩個是 constant function 有兩個是 balanced function，在不做任何取值的情況下盲猜的結果有 $1/2$ 的機會能猜中 f 是哪一類函數。

§1.2 Deutsch 演算法

Deutsch 演算法: 給定 $f\colon \{0,1\} \to \{0,1\}$。對於前述的 $\mathrm{U}_{f,\pm}$ 的機制輸入 $|+\rangle \equiv \mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$，再讓所出現的量子態通過一個 Hadamard 閘操作後進行測量，我們發現

1. 先將疊加態 $|+\rangle$ 送入 $\mathrm{U}_{f,\pm}$（它得是線性的才有下述性質）：
$$\mathrm{U}_{f,\pm}|+\rangle = \frac{1}{\sqrt{2}}\left[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle)\right].$$
2. 若 $f(0) = f(1)$，則上式為 $\pm|+\rangle$，則再經過一個 Hadamard 閘操作後會得到 $\pm|0\rangle$，此時再進行測量必得到 0。
3. 若 $f(0) \neq f(1)$，則上式為 $\pm|-\rangle$，則再經過一個 Hadamard 閘操作後會得到 $\pm|1\rangle$，此時再進行測量必得到 1。

因此，測量 $\mathrm{HU}_{f,\pm}\mathrm{H}|0\rangle$，若得到 0 則 f 為 constant function 若不為 0 則 f 為 balanced function。注意到在此只用到一個 query。

§1.2 Deutsch 演算法

Deutsch 演算法: 給定 $f\colon \{0,1\} \to \{0,1\}$。對於前述的 $\mathrm{U}_{f,\pm}$ 的機制輸入 $|+\rangle \equiv \mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$，再讓所出現的量子態通過一個 Hadamard 閘操作後進行測量，我們發現

1. 先將疊加態 $|+\rangle$ 送入 $\mathrm{U}_{f,\pm}$（它得是線性的才有下述性質）：
$$\mathrm{U}_{f,\pm}|+\rangle = \frac{1}{\sqrt{2}}\left[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle)\right].$$
2. 若 $f(0) = f(1)$，則上式為 $\pm|+\rangle$，則再經過一個 Hadamard 閘操作後會得到 $\pm|0\rangle$，此時再進行測量必得到 0。
3. 若 $f(0) \neq f(1)$，則上式為 $\pm|-\rangle$，則再經過一個 Hadamard 閘操作後會得到 $\pm|1\rangle$，此時再進行測量必得到 1。

因此，測量 $\mathrm{HU}_{f,\pm}\mathrm{H}|0\rangle$，若得到 0 則 f 為 constant function 若不為 0 則 f 為 balanced function。注意到在此只用到一個 query。

§1.2 Deutsch 演算法

Deutsch 演算法: 給定 $f\colon \{0,1\} \to \{0,1\}$。對於前述的 $\mathrm{U}_{f,\pm}$ 的機制輸入 $|+\rangle \equiv \mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$，再讓所出現的量子態通過一個 Hadamard 閘操作後進行測量，我們發現

1. 先將疊加態 $|+\rangle$ 送入 $\mathrm{U}_{f,\pm}$（它得是線性的才有下述性質）：
$$\mathrm{U}_{f,\pm}|+\rangle = \frac{1}{\sqrt{2}}\left[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle)\right].$$
2. 若 $f(0) = f(1)$，則上式為 $\pm|+\rangle$，則再經過一個 Hadamard 閘操作後會得到 $\pm|0\rangle$，此時再進行測量必得到 0。
3. 若 $f(0) \neq f(1)$，則上式為 $\pm|-\rangle$，則再經過一個 Hadamard 閘操作後會得到 $\pm|1\rangle$，此時再進行測量必得到 1。

因此，測量 $\mathrm{HU}_{f,\pm}\mathrm{H}|0\rangle$，若得到 0 則 f 為 constant function 若不為 0 則 f 為 balanced function。注意到在此只用到一個 query。

§1.2 Deutsch 演算法

Deutsch 演算法: 給定 $f\colon \{0,1\} \to \{0,1\}$。對於前述的 $\mathrm{U}_{f,\pm}$ 的機制輸入 $|+\rangle \equiv \mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$，再讓所出現的量子態通過一個 Hadamard 閘操作後進行測量，我們發現

1. 先將疊加態 $|+\rangle$ 送入 $\mathrm{U}_{f,\pm}$（它得是線性的才有下述性質）:
$$\mathrm{U}_{f,\pm}|+\rangle = \frac{1}{\sqrt{2}}\left[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle)\right].$$
2. 若 $f(0) = f(1)$，則上式為 $\pm|+\rangle$，則再經過一個 Hadamard 閘操作後會得到 $\pm|0\rangle$，此時再進行測量必得到 0。
3. 若 $f(0) \neq f(1)$，則上式為 $\pm|-\rangle$，則再經過一個 Hadamard 閘操作後會得到 $\pm|1\rangle$，此時再進行測量必得到 1。

因此，測量 $\mathrm{HU}_{f,\pm}\mathrm{H}|0\rangle$，若得到 0 則 f 為 constant function 若不為 0 則 f 為 balanced function。注意到在此只用到一個 query。

§1.2 Deutsch 演算法

Deutsch 演算法:

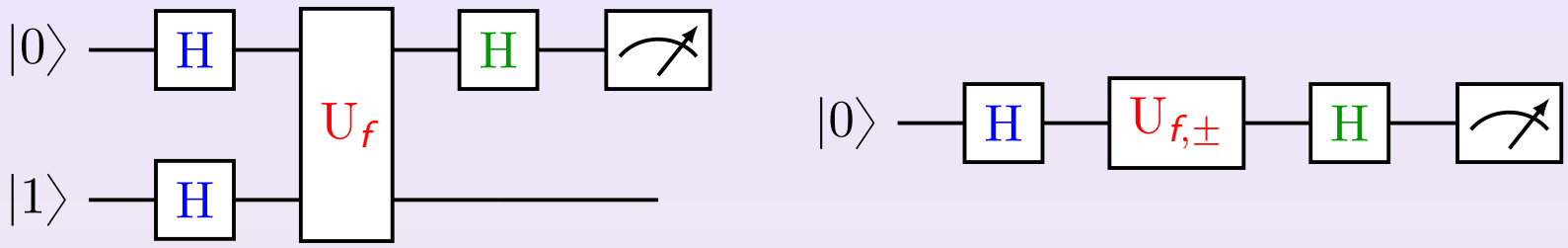


Figure 6: Deutsch 演算法之量子電路

§2.1 Deutsch-Jozsa 問題

Deutsch-Jozsa 問題: 假設有函數 $f: \{0,1\}^n \to \{0,1\}$ 滿足下列二者之一：

1. 對所有的 $x \in \{0,1\}^n$ f 取值相同，意即 f 為常數函數（此類的 f 被稱為 "**constant**" function）；
2. 在 $\{0,1\}^n$ 中有一半的 x 其函數值為 0 而另一半的 x 函數值為 1（此類的 f 被稱為 "**balanced**" function）。

Deutsch-Jozsa 問題的目標是找出 f 是 constant 或是 balanced。

注意：不似 $n = 1$（Deutsch 問題）的情況，在 $n > 1$ 的情形下函數 $f: \{0,1\}^n \to \{0,1\}$ 除了可能是 constant 或是 balanced 外，還可能有其它狀況（例如 $\{0,1\}^n$ 中的 1/4 函數值為 0 而 3/4 的函數值為 1）。因此，在 Deutsch-Jozsa 問題中的 f 被限定為 constant 或是 balanced 二者之一而無其它的可能性。

§2.1 Deutsch-Jozsa 問題

Deutsch-Jozsa 問題: 假設有函數 $f\colon \{0,1\}^n \to \{0,1\}$ 滿足下列二者之一：

1. 對所有的 $x \in \{0,1\}^n$ f 取值相同，意即 f 為常數函數（此類的 f 被稱為 "**constant**" function）；
2. 在 $\{0,1\}^n$ 中有一半的 x 其函數值為 0 而另一半的 x 函數值為 1（此類的 f 被稱為 "**balanced**" function）。

Deutsch-Jozsa 問題的目標是找出 f 是 constant 或是 balanced。

注意：不似 $n = 1$（Deutsch 問題）的情況，在 $n > 1$ 的情形下函數 $f\colon \{0,1\}^n \to \{0,1\}$ 除了可能是 constant 或是 balanced 外，還可能有其它狀況（例如 $\{0,1\}^n$ 中的 $1/4$ 函數值為 0 而 $3/4$ 的函數值為 1）。因此，在 Deutsch-Jozsa 問題中的 f 被限定為 constant 或是 balanced 二者之一而無其它的可能性。

§2.1 Deutsch-Jozsa 問題

傳統演算法的 query 次數估計：

任何的古典演算法（包含隨機演算法），最好的情況是 query 函數值兩次之後就知道 f 是 balanced function（兩次取值不同）；最差的情況是 query 函數值 $2^{n-1}+1$ 次才知道其為 balanced（前 2^{n-1} 次都取到同一個值，而第 $2^{n-1}+1$ 次取值取到不同值）或是 constant（這 $2^{n-1}+1$ 次都取到同一個函數值）。

隨機演算法：

由於只有兩個 constant functions 但是有 $2C_{2^{n-1}}^{2^n}$ 個 balanced functions，所以在 n 大的情況下即使不做任何 querry 猜 f 是 balanced function 時猜中的機率很高。

§2.1 Deutsch-Jozsa 問題

傳統演算法的 query 次數估計：

任何的古典演算法（包含隨機演算法），最好的情況是 query 函數值兩次之後就知道 f 是 balanced function（兩次取值不同）；最差的情況是 query 函數值 $2^{n-1}+1$ 次才知道其為 balanced（前 2^{n-1} 次都取到同一個值，而第 $2^{n-1}+1$ 次取值取到不同值）或是 constant（這 $2^{n-1}+1$ 次都取到同一個函數值）。

隨機演算法：

由於只有兩個 constant functions 但是有 $2C_{2^{n-1}}^{2^n}$ 個 balanced functions，所以在 n 大的情況下即使不做任何 querry 猜 f 是 balanced function 時猜中的機率很高。

§2.2 Deutsch-Jozsa 演算法

Deutsch-Jozsa 演算法: 給定不是 constant 就是 balanced 的函數 $f\colon \{0,1\}^n \to \{0,1\}$。將 $\mathrm{H}^{\otimes n}|0\rangle^{\otimes n}$ 代入前述的 $\mathrm{U}_{f,\pm}$ 機制，再讓所出現的量子態通過 $\mathrm{H}^{\otimes n}$ 閘操作，最後對最終量子態的 n 個量子位元進行測量。

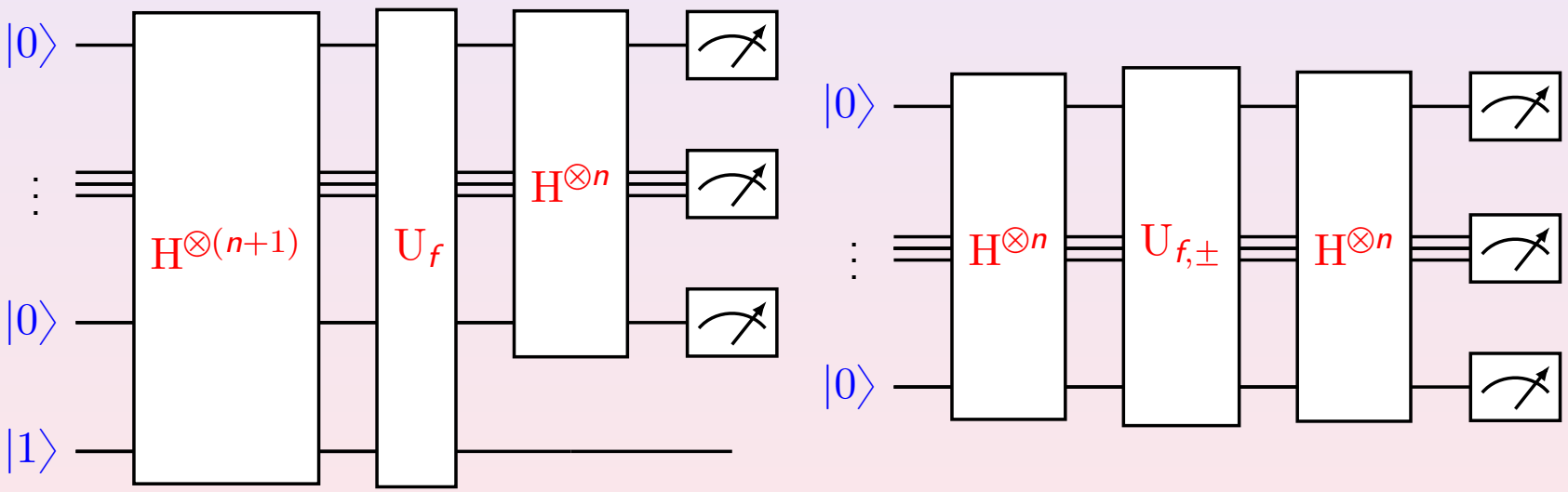


Figure 7: Deutsch-Jozsa 演算法的量子電路

§2.2 Deutsch-Jozsa 演算法

按步驟進行 Deutsch-Jozsa 演算法，我們發現

1. $\mathrm{H}^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle$，在此提醒大家若 $j = (j_1 j_2 \cdots j_n)_2$ 則以 $|j\rangle$ 表示 $|j_1 j_2 \cdots j_n\rangle$。
2. 將 $\mathrm{H}^{\otimes n}|0\rangle^{\otimes n}$ 代入 $\mathrm{U}_{f,\pm}$ 得到
$$\mathrm{U}_{f,\pm}\left(\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)}|j\rangle.$$
3. 將 $\mathrm{H}^{\otimes n}$ 作用於 ② 中的結果，我們得到
$$\mathrm{H}^{\otimes n}\left(\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)}|j\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)}\mathrm{H}^{\otimes n}|j\rangle.$$

接下來我們推導 ① 與 ③ 中都有出現的 $\mathrm{H}^{\otimes n}|j\rangle$。

§2.2 Deutsch-Jozsa 演算法

按步驟進行 Deutsch-Jozsa 演算法，我們發現

1. $\mathrm{H}^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle$，在此提醒大家若 $j = (j_1 j_2 \cdots j_n)_2$ 則以 $|j\rangle$ 表示 $|j_1 j_2 \cdots j_n\rangle$。
2. 將 $\mathrm{H}^{\otimes n}|0\rangle^{\otimes n}$ 代入 $\mathrm{U}_{f,\pm}$ 得到

$$\mathrm{U}_{f,\pm}\left(\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)}|j\rangle.$$

3. 將 $\mathrm{H}^{\otimes n}$ 作用於 ② 中的結果，我們得到

$$\mathrm{H}^{\otimes n}\left(\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)}|j\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)}\mathrm{H}^{\otimes n}|j\rangle.$$

接下來我們推導 ① 與 ③ 中都有出現的 $\mathrm{H}^{\otimes n}|j\rangle$。

§2.2 Deutsch-Jozsa 演算法

按步驟進行 Deutsch-Jozsa 演算法，我們發現

1. $\mathrm{H}^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle$，在此提醒大家若 $j = (j_1 j_2 \cdots j_n)_2$

   則以 $|j\rangle$ 表示 $|j_1 j_2 \cdots j_n\rangle$。

2. 將 $\mathrm{H}^{\otimes n}|0\rangle^{\otimes n}$ 代入 $\mathrm{U}_{f,\pm}$ 得到

$$\mathrm{U}_{f,\pm}\left(\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)}|j\rangle.$$

3. 將 $\mathrm{H}^{\otimes n}$ 作用於 ② 中的結果，我們得到

$$\mathrm{H}^{\otimes n}\left(\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)}|j\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)}\mathrm{H}^{\otimes n}|j\rangle.$$

接下來我們推導 ① 與 ③ 中都有出現的 $\mathrm{H}^{\otimes n}|j\rangle$。

§2.2 Deutsch-Jozsa 演算法

按步驟進行 Deutsch-Jozsa 演算法，我們發現

1. $\mathrm{H}^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle$，在此提醒大家若 $j = (j_1 j_2 \cdots j_n)_2$

   則以 $|j\rangle$ 表示 $|j_1 j_2 \cdots j_n\rangle$。

2. 將 $\mathrm{H}^{\otimes n}|0\rangle^{\otimes n}$ 代入 $\mathrm{U}_{f,\pm}$ 得到

$$\mathrm{U}_{f,\pm}\left(\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)}|j\rangle.$$

3. 將 $\mathrm{H}^{\otimes n}$ 作用於 ② 中的結果，我們得到

$$\mathrm{H}^{\otimes n}\left(\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)}|j\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)}\mathrm{H}^{\otimes n}|j\rangle.$$

接下來我們推導 ① 與 ③ 中都有出現的 $\mathrm{H}^{\otimes n}|j\rangle$。

§2.2 Deutsch-Jozsa 演算法

Theorem

For each $n \in \mathbb{N}$ and $j = (j_1 j_2 \cdots j_n)_2$,

$$\mathrm{H}^{\otimes n}|j\rangle \equiv \mathrm{H}^{\otimes n}|j_1 j_2 \cdots j_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} (-1)^{j \bullet k} |k\rangle, \tag{1}$$

where $j \bullet k \equiv j_1 k_1 + \cdots j_n k_n$ if $k = (k_1 k_2 \cdots k_n)_2$.

Proof.

Note that for $j_\ell \in \{0,1\}$, $\mathrm{H}|j_\ell\rangle = \frac{1}{\sqrt{2}} \sum_{k_\ell=0}^{1} (-1)^{j_\ell k_\ell} |k_\ell\rangle$. Therefore,

$$\begin{aligned}
\mathrm{H}^{\otimes n}|j_1 j_2 \cdots j_n\rangle &\equiv (\mathrm{H}|j_1\rangle) \otimes \cdots \otimes (\mathrm{H}|j_n\rangle) \\
&= \left(\frac{1}{\sqrt{2}} \sum_{k_1=0}^{1} (-1)^{j_1 k_1} |k_1\rangle\right) \otimes \cdots \otimes \left(\frac{1}{\sqrt{2}} \sum_{k_n=0}^{1} (-1)^{j_n k_n} |k_n\rangle\right) \\
&= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} (-1)^{j_1 k_1 + \cdots + j_n k_n} |k_1\rangle \otimes \cdots \otimes |k_n\rangle .
\end{aligned}$$

□

§2.2 Deutsch-Jozsa 演算法

Theorem

For each $n \in \mathbb{N}$ *and* $j = (j_1 j_2 \cdots j_n)_2$,

$$\mathrm{H}^{\otimes n}|j\rangle \equiv \mathrm{H}^{\otimes n}|j_1 j_2 \cdots j_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} (-1)^{j \bullet k} |k\rangle, \tag{1}$$

where $j \bullet k \equiv j_1 k_1 + \cdots j_n k_n$ *if* $k = (k_1 k_2 \cdots k_n)_2$.

Proof.

Note that for $j_\ell \in \{0, 1\}$, $\mathrm{H}|j_\ell\rangle = \frac{1}{\sqrt{2}} \sum_{k_\ell=0}^{1} (-1)^{j_\ell k_\ell} |k_\ell\rangle$. Therefore,

$$\begin{aligned}
\mathrm{H}^{\otimes n}|j_1 j_2 \cdots j_n\rangle &\equiv (\mathrm{H}|j_1\rangle) \otimes \cdots \otimes (\mathrm{H}|j_n\rangle) \\
&= \left(\frac{1}{\sqrt{2}} \sum_{k_1=0}^{1} (-1)^{j_1 k_1} |k_1\rangle\right) \otimes \cdots \otimes \left(\frac{1}{\sqrt{2}} \sum_{k_n=0}^{1} (-1)^{j_n k_n} |k_n\rangle\right) \\
&= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} (-1)^{j_1 k_1 + \cdots + j_n k_n} |k_1\rangle \otimes \cdots \otimes |k_n\rangle.
\end{aligned}$$

□

§2.2 Deutsch-Jozsa 演算法

Theorem

For each $n \in \mathbb{N}$ *and* $j = (j_1 j_2 \cdots j_n)_2$,

$$\mathrm{H}^{\otimes n}|j\rangle \equiv \mathrm{H}^{\otimes n}|j_1 j_2 \cdots j_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} (-1)^{j \bullet k} |k\rangle, \tag{1}$$

where $j \bullet k \equiv j_1 k_1 + \cdots j_n k_n$ *if* $k = (k_1 k_2 \cdots k_n)_2$.

Proof.

Note that for $j_\ell \in \{0, 1\}$, $\mathrm{H}|j_\ell\rangle = \frac{1}{\sqrt{2}} \sum_{k_\ell=0}^{1} (-1)^{j_\ell k_\ell} |k_\ell\rangle$. Therefore,

$$\begin{aligned}
\mathrm{H}^{\otimes n}|j_1 j_2 \cdots j_n\rangle &\equiv (\mathrm{H}|j_1\rangle) \otimes \cdots \otimes (\mathrm{H}|j_n\rangle) \\
&= \left(\frac{1}{\sqrt{2}} \sum_{k_1=0}^{1} (-1)^{j_1 k_1} |k_1\rangle\right) \otimes \cdots \otimes \left(\frac{1}{\sqrt{2}} \sum_{k_n=0}^{1} (-1)^{j_n k_n} |k_n\rangle\right) \\
&= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} (-1)^{j_1 k_1 + \cdots + j_n k_n} |k_1\rangle \otimes \cdots \otimes |k_n\rangle.
\end{aligned}$$

□

§2.2 Deutsch-Jozsa 演算法

Theorem

For each $n \in \mathbb{N}$ *and* $j = (j_1 j_2 \cdots j_n)_2$,

$$\mathrm{H}^{\otimes n}|j\rangle \equiv \mathrm{H}^{\otimes n}|j_1 j_2 \cdots j_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} (-1)^{j \bullet k} |k\rangle, \tag{1}$$

where $j \bullet k \equiv j_1 k_1 + \cdots j_n k_n$ *if* $k = (k_1 k_2 \cdots k_n)_2$.

Proof.

Note that for $j_\ell \in \{0,1\}$, $\mathrm{H}|j_\ell\rangle = \frac{1}{\sqrt{2}} \sum_{k_\ell=0}^{1} (-1)^{j_\ell k_\ell} |k_\ell\rangle$. Therefore,

$$\begin{aligned}
\mathrm{H}^{\otimes n}|j_1 j_2 \cdots j_n\rangle &\equiv (\mathrm{H}|j_1\rangle) \otimes \cdots \otimes (\mathrm{H}|j_n\rangle) \\
&= \left(\frac{1}{\sqrt{2}} \sum_{k_1=0}^{1} (-1)^{j_1 k_1} |k_1\rangle\right) \otimes \cdots \otimes \left(\frac{1}{\sqrt{2}} \sum_{k_n=0}^{1} (-1)^{j_n k_n} |k_n\rangle\right) \\
&= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} (-1)^{j_1 k_1 + \cdots + j_n k_n} |k_1\rangle \otimes \cdots \otimes |k_n\rangle. \qquad \square
\end{aligned}$$

§2.2 Deutsch-Jozsa 演算法

Theorem

For each $n \in \mathbb{N}$ *and* $j = (j_1 j_2 \cdots j_n)_2$,

$$\mathrm{H}^{\otimes n}|j\rangle \equiv \mathrm{H}^{\otimes n}|j_1 j_2 \cdots j_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} (-1)^{j \bullet k} |k\rangle, \qquad (1)$$

where $j \bullet k \equiv j_1 k_1 + \cdots j_n k_n$ *if* $k = (k_1 k_2 \cdots k_n)_2$.

Proof.

Note that for $j_\ell \in \{0,1\}$, $\mathrm{H}|j_\ell\rangle = \frac{1}{\sqrt{2}} \sum_{k_\ell=0}^{1} (-1)^{j_\ell k_\ell} |k_\ell\rangle$. Therefore,

$$\begin{aligned}
\mathrm{H}^{\otimes n}|j_1 j_2 \cdots j_n\rangle &\equiv (\mathrm{H}|j_1\rangle) \otimes \cdots \otimes (\mathrm{H}|j_n\rangle) \\
&= \left(\frac{1}{\sqrt{2}} \sum_{k_1=0}^{1} (-1)^{j_1 k_1} |k_1\rangle\right) \otimes \cdots \otimes \left(\frac{1}{\sqrt{2}} \sum_{k_n=0}^{1} (-1)^{j_n k_n} |k_n\rangle\right) \\
&= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} (-1)^{j_1 k_1 + \cdots + j_n k_n} |k_1\rangle \otimes \cdots \otimes |k_n\rangle .
\end{aligned}$$

□

§2.2 Deutsch-Jozsa 演算法

經過上述步驟「將 $\mathrm{H}^{\otimes n}|0\rangle^{\otimes n}$ 代入 $\mathrm{U}_{f,\pm}$，接著將 $\mathrm{H}^{\otimes n}$ 作用於所得量子態」，使用 (1) 式我們得到

$$\mathrm{H}^{\otimes n}\mathrm{U}_{f,\pm}\mathrm{H}^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}(-1)^{f(j)}\mathrm{H}^{\otimes n}|j\rangle$$

$$= \frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}(-1)^{f(j)}\left(\frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}(-1)^{j\bullet k}|k\rangle\right).$$

最後的疊加態中 $|0\rangle^{\otimes n}$ 的機率振幅為 $\frac{1}{2^n}\sum_{j=0}^{2^n-1}(-1)^{f(j)}$。

1. 若 f 為 constant function，則 $|0\rangle^{\otimes n}$ 的機率振幅為 ± 1，因此測量此最後量子態得到 $|0\rangle^{\otimes n}$ 之機率為 1。
2. 若 f 為 balanced function，則 $|0\rangle^{\otimes n}$ 的機率振幅為 0，因此測量此最後量子態不會得到 $|0\rangle^{\otimes n}$ 之機率為 1。

§2.2 Deutsch-Jozsa 演算法

經過上述步驟「將 $\mathrm{H}^{\otimes n}|0\rangle^{\otimes n}$ 代入 $\mathrm{U}_{f,\pm}$，接著將 $\mathrm{H}^{\otimes n}$ 作用於所得量子態」，使用 (1) 式我們得到

$$\mathrm{H}^{\otimes n}\mathrm{U}_{f,\pm}\mathrm{H}^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}(-1)^{f(j)}\mathrm{H}^{\otimes n}|j\rangle$$

$$= \frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}(-1)^{f(j)}\left(\frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}(-1)^{j\bullet k}|k\rangle\right).$$

最後的疊加態中 $|0\rangle^{\otimes n}$ 的機率振幅為 $\frac{1}{2^n}\sum_{j=0}^{2^n-1}(-1)^{f(j)}$。

1. 若 f 為 constant function，則 $|0\rangle^{\otimes n}$ 的機率振幅為 ± 1，因此測量此最後量子態得到 $|0\rangle^{\otimes n}$ 之機率為 1。
2. 若 f 為 balanced function，則 $|0\rangle^{\otimes n}$ 的機率振幅為 0，因此測量此最後量子態不會得到 $|0\rangle^{\otimes n}$ 之機率為 1。

§2.2 Deutsch-Jozsa 演算法

因此 Deutsch-Jozsa 問題可用 Deutsch-Jozsa 演算法在只進行一次 query 之後就解決（確知函數 f 是 constant 或是 balanced），代價是需要 $\mathcal{O}(n)$ 個量子邏輯閘（Hadamard 閘）。

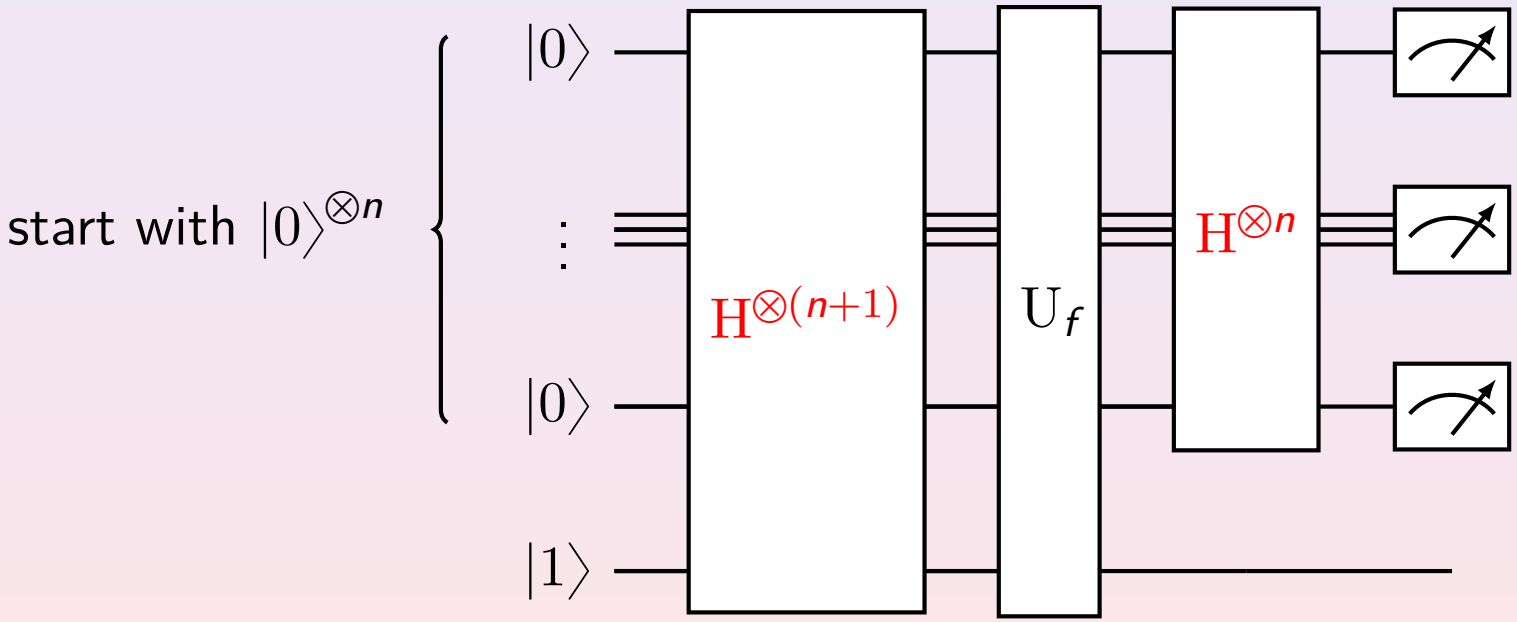


Figure 8: Deutsch-Jozsa 演算法的量子電路

§3.1 搜尋問題

搜尋問題：給定一函數 $f\colon \{0,1\}^N \to \{0,1\}$，其中 $N = 2^n$。搜尋問題的目標是找到一個 x 使得 $f(x) = 1$（或是當沒有這樣的 x 時輸出「無解」。）

1. 此問題可以被看成在 N 個不同未排序（無結構）的資料中搜尋所要的資料的簡化版問題。在此若一開始的資料數不是二的次方數的話，就直接補一堆無用的資料讓總資料數為二的次方數。
2. 古典的（隨機）演算法需要 $\mathcal{O}(N)$ 次的 queries 才能搜尋到滿足搜尋條件的資料，而 Grover 演算法只需要 $\mathcal{O}(\sqrt{N})$ 次的 **queries** 以及 $\mathcal{O}(\sqrt{N}\log_2 N)$ 個與取值無關的量子邏輯閘就能完成搜尋。

§3.1 搜尋問題

搜尋問題：給定一函數 $f\colon \{0,1\}^N \to \{0,1\}$，其中 $N = 2^n$。搜尋問題的目標是找到一個 x 使得 $f(x) = 1$（或是當沒有這樣的 x 時輸出「無解」。）

1. 此問題可以被看成在 N 個不同未排序（無結構）的資料中搜尋所要的資料的簡化版問題。在此若一開始的資料數不是二的次方數的話，就直接補一堆無用的資料讓總資料數為二的次方數。
2. 古典的（隨機）演算法需要 $\mathcal{O}(N)$ 次的 queries 才能搜尋到滿足搜尋條件的資料，而 Grover 演算法只需要 $\mathcal{O}(\sqrt{N})$ 次的 queries 以及 $\mathcal{O}(\sqrt{N}\log_2 N)$ 個與取值無關的量子邏輯閘就能完成搜尋。

§3.1 搜尋問題

搜尋問題：給定一函數 $f\colon \{0,1\}^N \to \{0,1\}$，其中 $N = 2^n$。搜尋問題的目標是找到一個 x 使得 $f(x) = 1$（或是當沒有這樣的 x 時輸出「無解」。）

1. 此問題可以被看成在 N 個不同未排序（無結構）的資料中搜尋所要的資料的簡化版問題。在此若一開始的資料數不是二的次方數的話，就直接補一堆無用的資料讓總資料數為二的次方數。
2. 古典的（隨機）演算法需要 $\mathcal{O}(N)$ 次的 queries 才能搜尋到滿足搜尋條件的資料，而 Grover 演算法只需要 $\mathcal{O}(\sqrt{N})$ 次的 **queries** 以及 $\mathcal{O}(\sqrt{N}\log_2 N)$ 個與取值無關的量子邏輯閘就能完成搜尋。

§3.2 Grover 演算法

在介紹 Grover 搜尋演算法前，首先我們介紹所謂的一個 Grover iterate。記得 $N = 2^n$。

1. 令 R 表示在所有非零的基底態 (basis state) $|i_1 \cdots i_n\rangle$ 係數加上負號但維持 $|0\rangle^{\otimes n}$ 前的係數的么正變換，意即

$$\mathrm{R}\Big(\sum_{j=0}^{N-1} a_j|j\rangle\Big) = a_0|0\rangle - \sum_{1\leqslant j\leqslant 2^n-1} a_j|j\rangle.$$

2. 給定一個搜尋判斷法則 $f\colon \{0,1\}^N \to \{0,1\}$，我們有上帝所給的 U_f 或 $\mathrm{U}_{f,\pm}$。

一個 Grover iterate 即 $\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}\mathrm{U}_{f,\pm}$，我們以 $\mathcal{G}$ 來代表一個 Grover iterate。注意到一個 Grover iterate 使用一個 query 以及 $\mathcal{O}(\log_2 N)$ 個量子邏輯閘。

§3.2 Grover 演算法

在介紹 Grover 搜尋演算法前，首先我們介紹所謂的一個 Grover iterate。記得 $N = 2^n$。

1. 令 R 表示在所有非零的基底態 (basis state) $|i_1 \cdots i_n\rangle$ 係數加上負號但維持 $|0\rangle^{\otimes n}$ 前的係數的么正變換，意即

$$\mathrm{R}\Big(\sum_{j=0}^{N-1} a_j|j\rangle\Big) = a_0|0\rangle - \sum_{1\leqslant j\leqslant 2^n-1} a_j|j\rangle.$$

2. 給定一個搜尋判斷法則 $f\colon \{0,1\}^N \to \{0,1\}$，我們有上帝所給的 U_f 或 $\mathrm{U}_{f,\pm}$。

一個 Grover iterate 即 $\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}\mathrm{U}_{f,\pm}$，我們以 $\mathcal{G}$ 來代表一個 Grover iterate。注意到一個 Grover iterate 使用一個 query 以及 $\mathcal{O}(\log_2 N)$ 個量子邏輯閘。

§3.2 Grover 演算法

在介紹 Grover 搜尋演算法前，首先我們介紹所謂的一個 Grover iterate。記得 $N = 2^n$。

1. 令 R 表示在所有非零的基底態 (basis state) $|i_1 \cdots i_n\rangle$ 係數加上負號但維持 $|0\rangle^{\otimes n}$ 前的係數的么正變換，意即

$$\mathrm{R}\Big(\sum_{j=0}^{N-1} a_j|j\rangle\Big) = a_0|0\rangle - \sum_{1\leqslant j\leqslant 2^n-1} a_j|j\rangle.$$

2. 給定一個搜尋判斷法則 $f\colon \{0,1\}^N \to \{0,1\}$，我們有上帝所給的 U_f 或 $\mathrm{U}_{f,\pm}$。

一個 Grover iterate 即 $\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}\mathrm{U}_{f,\pm}$，我們以 $\mathcal{G}$ 來代表一個 Grover iterate。注意到一個 Grover iterate 使用一個 query 以及 $\mathcal{O}(\log_2 N)$ 個量子邏輯閘。

§3.2 Grover 演算法

在介紹 Grover 搜尋演算法前，首先我們介紹所謂的一個 Grover iterate。記得 $N = 2^n$。

1. 令 R 表示在所有非零的基底態 (basis state) $|i_1 \cdots i_n\rangle$ 係數加上負號但維持 $|0\rangle^{\otimes n}$ 前的係數的么正變換，意即

$$\mathrm{R}\Big(\sum_{j=0}^{N-1} a_j|j\rangle\Big) = a_0|0\rangle - \sum_{1\leqslant j\leqslant 2^n-1} a_j|j\rangle.$$

2. 給定一個搜尋判斷法則 $f\colon \{0,1\}^N \to \{0,1\}$，我們有上帝所給的 U_f 或 $\mathrm{U}_{f,\pm}$。

一個 Grover iterate 即 $\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}\mathrm{U}_{f,\pm}$，我們以 $\mathcal{G}$ 來代表一個 Grover iterate。注意到一個 Grover iterate 使用一個 query 以及 $\mathcal{O}(\log_2 N)$ 個量子邏輯閘。

§3.2 Grover 演算法

在介紹 Grover 搜尋演算法前，首先我們介紹所謂的一個 Grover iterate。記得 $N = 2^n$。

1. 令 R 表示在所有非零的基底態 (basis state) $|i_1 \cdots i_n\rangle$ 係數加上負號但維持 $|0\rangle^{\otimes n}$ 前的係數的么正變換，意即

$$\mathrm{R}\Big(\sum_{j=0}^{N-1} a_j|j\rangle\Big) = a_0|0\rangle - \sum_{1\leqslant j\leqslant 2^n-1} a_j|j\rangle.$$

2. 給定一個搜尋判斷法則 $f\colon \{0,1\}^N \to \{0,1\}$，我們有上帝所給的 U_f 或 $\mathrm{U}_{f,\pm}$。

一個 Grover iterate 即 $\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}\mathrm{U}_{f,\pm}$，我們以 $\mathcal{G}$ 來代表一個 Grover iterate。注意到一個 Grover iterate 使用一個 query 以及 $\mathcal{O}(\log_2 N)$ 個量子邏輯閘。

§3.2 Grover 演算法

以下的量子電路圖為 Grover 搜尋演算法的量子電路。

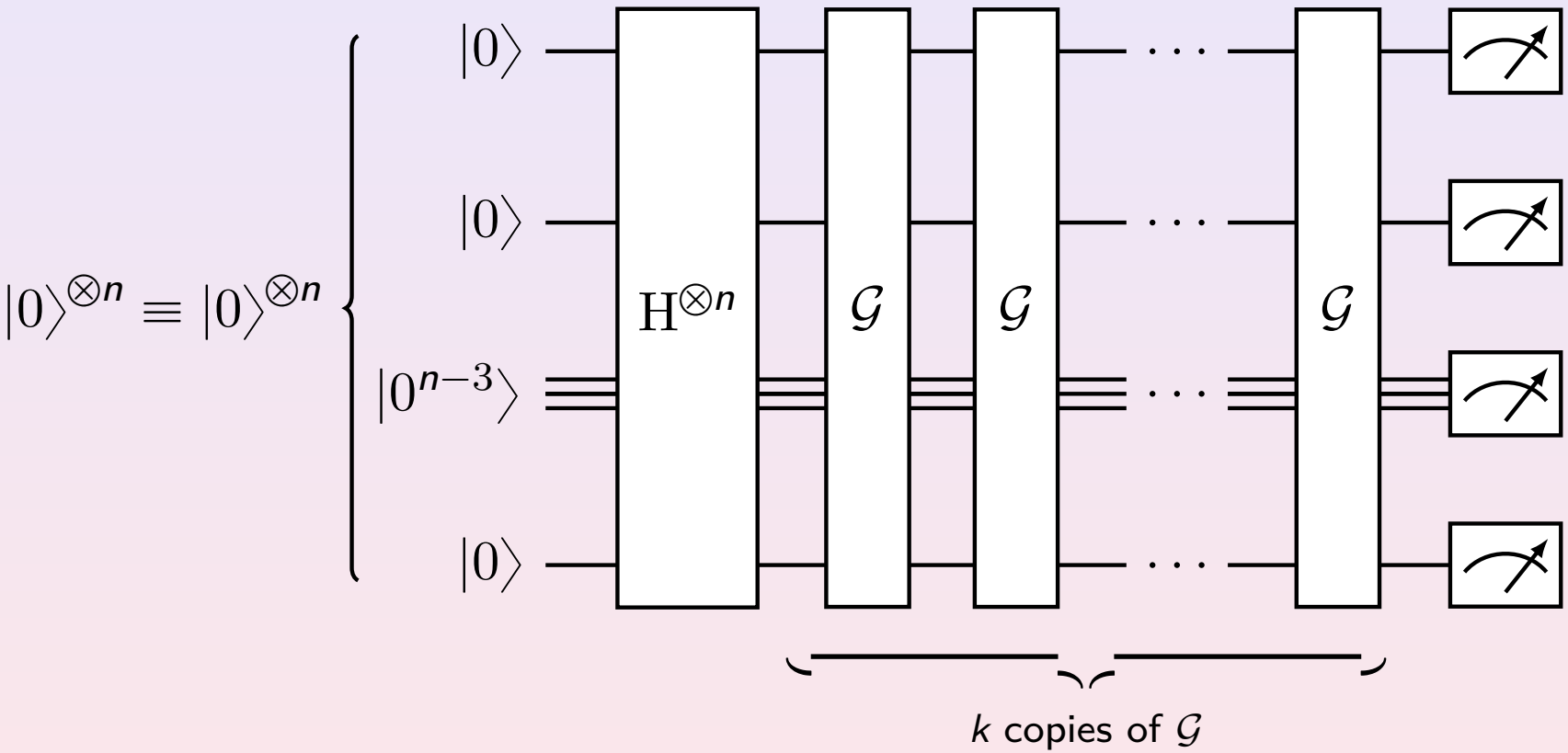


Figure 9: 具有 k 個 Grover iterates 的 Grover 演算法

§3.2 Grover 演算法

要分析 Grover 演算法的作用，我們首先引進所謂的 "good" states $|G\rangle$ 和 "bad" states $|B\rangle$：

$$|G\rangle = \frac{1}{\sqrt{t}} \sum_{\{j|\, f(j)=1\}} |j\rangle \qquad \text{and} \qquad |B\rangle = \frac{1}{\sqrt{N-t}} \sum_{\{j|\, f(j)=0\}} |j\rangle .$$

其中 $t = \#\{j \,|\, f(j) = 1\}$ 代表 f 的 Hamming weight。那麼所有基底態的均勻疊加態 $|U\rangle \equiv \mathrm{H}^{\otimes n}|0\rangle^{\otimes n}$ 可以寫成 $|G\rangle$ 與 $|B\rangle$ 的線性組合：

$$|U\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle = \frac{1}{\sqrt{N}} \Big(\sum_{\{j|\, f(j)=1\}} + \sum_{\{j|\, f(j)=0\}} \Big) |j\rangle$$

$$= \frac{1}{\sqrt{N}} \Big(\sqrt{t}\,|G\rangle + \sqrt{N-t}\,|B\rangle \Big) = \sin\theta |G\rangle + \cos\theta |B\rangle ,$$

在此 $\theta = \arcsin \sqrt{\frac{t}{N}}$。

§3.2 Grover 演算法

要分析 Grover 演算法的作用，我們首先引進所謂的 "good" states $|G\rangle$ 和 "bad" states $|B\rangle$：

$$|G\rangle = \frac{1}{\sqrt{t}} \sum_{\{j|\, f(j)=1\}} |j\rangle \qquad \text{and} \qquad |B\rangle = \frac{1}{\sqrt{N-t}} \sum_{\{j|\, f(j)=0\}} |j\rangle .$$

其中 $t = \#\{j \,|\, f(j) = 1\}$ 代表 f 的 Hamming weight。那麼所有基底態的均勻疊加態 $|U\rangle \equiv \mathrm{H}^{\otimes n}|0\rangle^{\otimes n}$ 可以寫成 $|G\rangle$ 與 $|B\rangle$ 的線性組合：

$$\begin{aligned} |U\rangle &= \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle = \frac{1}{\sqrt{N}} \Big(\sum_{\{j|\, f(j)=1\}} + \sum_{\{j|\, f(j)=0\}} \Big) |j\rangle \\ &= \frac{1}{\sqrt{N}} \Big(\sqrt{t}\,|G\rangle + \sqrt{N-t}\,|B\rangle \Big) = \sin\theta |G\rangle + \cos\theta |B\rangle , \end{aligned}$$

在此 $\theta = \arcsin \sqrt{\dfrac{t}{N}}$。

§3.2 Grover 演算法

令 $\mathbb{P}$ 為由 $|G\rangle$ 與 $|B\rangle$ 所生成的線性空間（或說 $\mathbb{P}$ 包含了所有形如 $a|G\rangle + b|B\rangle$ 的向量）。則當將 Grover iterate $\mathcal{G}$ 視為只作用在 $\mathbb{P}$ 上的一個算子，$\mathcal{G}$ 的作用事實上是依序對 $|B\rangle$ 與 $|U\rangle$ 做鏡射。

- 什麼叫做一個向量 u 對另一個非零向量 v 做鏡射？

答案：給定一非零向量 v。任一向量 u 永遠可以拆解成平行於 v 與垂直於 v 的兩個分量。若將 u 平行於 v 的分量叫 $u_{/\!/}$ 以及將 u 垂直於 v 的分量叫 $u_{\perp}$（在此所提到垂直概念一定與某個內積有關），則所謂的 u 對 v 做鏡射是指 $u \mapsto u_{/\!/} - u_{\perp}$ 這個操作。

- 一個向量 u 對單位向量 v 做鏡射的數學操作：假設 $\langle \cdot, \cdot \rangle$ 為垂直概念所依賴的內積，則 $u_{/\!/} = \langle v, u \rangle v$ 且因此 $u_{\perp} = u - \langle v, u \rangle v$。所以 u 對單位向量 v 做鏡射得到 $2\langle v, u \rangle v - u$。此鏡射操作為對 u 的線性變換。

§3.2 Grover 演算法

令 $\mathbb{P}$ 為由 $|G\rangle$ 與 $|B\rangle$ 所生成的線性空間（或說 $\mathbb{P}$ 包含了所有形如 $a|G\rangle + b|B\rangle$ 的向量）。則當將 Grover iterate $\mathcal{G}$ 視為只作用在 $\mathbb{P}$ 上的一個算子，$\mathcal{G}$ 的作用事實上是依序對 $|B\rangle$ 與 $|U\rangle$ 做鏡射。

- 什麼叫做一個向量 $\boldsymbol{u}$ 對另一個非零向量 $\boldsymbol{v}$ 做鏡射？

答案：給定一非零向量 $\boldsymbol{v}$。任一向量 $\boldsymbol{u}$ 永遠可以拆解成平行於 $\boldsymbol{v}$ 與垂直於 $\boldsymbol{v}$ 的兩個分量。若將 $\boldsymbol{u}$ 平行於 $\boldsymbol{v}$ 的分量叫 $\boldsymbol{u}_{/\!/}$ 以及將 $\boldsymbol{u}$ 垂直於 $\boldsymbol{v}$ 的分量叫 $\boldsymbol{u}_{\perp}$（在此所提到垂直概念一定與某個內積有關），則所謂的 $\boldsymbol{u}$ 對 $\boldsymbol{v}$ 做鏡射是指 $\boldsymbol{u} \mapsto \boldsymbol{u}_{/\!/} - \boldsymbol{u}_{\perp}$ 這個操作。

- 一個向量 $\boldsymbol{u}$ 對單位向量 $\boldsymbol{v}$ 做鏡射的數學操作：假設 $\langle \cdot, \cdot \rangle$ 為垂直概念所依賴的內積，則 $\boldsymbol{u}_{/\!/} = \langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{v}$ 且因此 $\boldsymbol{u}_{\perp} = \boldsymbol{u} - \langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{v}$。所以 $\boldsymbol{u}$ 對單位向量 $\boldsymbol{v}$ 做鏡射得到 $2\langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{v} - \boldsymbol{u}$。此鏡射操作為對 $\boldsymbol{u}$ 的線性變換。

§3.2 Grover 演算法

令 $\mathbb{P}$ 為由 $|G\rangle$ 與 $|B\rangle$ 所生成的線性空間（或說 $\mathbb{P}$ 包含了所有形如 $a|G\rangle + b|B\rangle$ 的向量）。則當將 Grover iterate $\mathcal{G}$ 視為只作用在 $\mathbb{P}$ 上的一個算子，$\mathcal{G}$ 的作用事實上是依序對 $|B\rangle$ 與 $|U\rangle$ 做鏡射。

- 什麼叫做一個向量 $\boldsymbol{u}$ 對另一個非零向量 $\boldsymbol{v}$ 做鏡射？

答案：給定一非零向量 $\boldsymbol{v}$。任一向量 $\boldsymbol{u}$ 永遠可以拆解成平行於 $\boldsymbol{v}$ 與垂直於 $\boldsymbol{v}$ 的兩個分量。若將 $\boldsymbol{u}$ 平行於 $\boldsymbol{v}$ 的分量叫 $\boldsymbol{u}_{/\!/}$ 以及將 $\boldsymbol{u}$ 垂直於 $\boldsymbol{v}$ 的分量叫 $\boldsymbol{u}_{\perp}$（在此所提到垂直概念一定與某個內積有關），則所謂的 $\boldsymbol{u}$ 對 $\boldsymbol{v}$ 做鏡射是指 $\boldsymbol{u} \mapsto \boldsymbol{u}_{/\!/} - \boldsymbol{u}_{\perp}$ 這個操作。

- 一個向量 $\boldsymbol{u}$ 對單位向量 $\boldsymbol{v}$ 做鏡射的數學操作：假設 $\langle \cdot, \cdot \rangle$ 為垂直概念所依賴的內積，則 $\boldsymbol{u}_{/\!/} = \langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{v}$ 且因此 $\boldsymbol{u}_{\perp} = \boldsymbol{u} - \langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{v}$。所以 $\boldsymbol{u}$ 對單位向量 $\boldsymbol{v}$ 做鏡射得到 $2\langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{v} - \boldsymbol{u}$。此鏡射操作為對 $\boldsymbol{u}$ 的線性變換。

§3.2 Grover 演算法

令 $\mathbb{P}$ 為由 $|G\rangle$ 與 $|B\rangle$ 所生成的線性空間（或說 $\mathbb{P}$ 包含了所有形如 $a|G\rangle + b|B\rangle$ 的向量）。則當將 Grover iterate $\mathcal{G}$ 視為只作用在 $\mathbb{P}$ 上的一個算子，$\mathcal{G}$ 的作用事實上是依序對 $|B\rangle$ 與 $|U\rangle$ 做鏡射。

- 什麼叫做一個向量 $\boldsymbol{u}$ 對另一個非零向量 $\boldsymbol{v}$ 做鏡射？

答案：給定一非零向量 $\boldsymbol{v}$。任一向量 $\boldsymbol{u}$ 永遠可以拆解成平行於 $\boldsymbol{v}$ 與垂直於 $\boldsymbol{v}$ 的兩個分量。若將 $\boldsymbol{u}$ 平行於 $\boldsymbol{v}$ 的分量叫 $\boldsymbol{u}_{/\!/}$ 以及將 $\boldsymbol{u}$ 垂直於 $\boldsymbol{v}$ 的分量叫 $\boldsymbol{u}_{\perp}$（在此所提到垂直概念一定與某個內積有關），則所謂的 $\boldsymbol{u}$ 對 $\boldsymbol{v}$ 做鏡射是指 $\boldsymbol{u} \mapsto \boldsymbol{u}_{/\!/} - \boldsymbol{u}_{\perp}$ 這個操作。

- 一個向量 $\boldsymbol{u}$ 對單位向量 $\boldsymbol{v}$ 做鏡射的數學操作：假設 $\langle \cdot, \cdot \rangle$ 為垂直概念所依賴的內積，則 $\boldsymbol{u}_{/\!/} = \langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{v}$ 且因此 $\boldsymbol{u}_{\perp} = \boldsymbol{u} - \langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{v}$。所以 $\boldsymbol{u}$ 對單位向量 $\boldsymbol{v}$ 做鏡射得到 $2\langle \boldsymbol{v}, \boldsymbol{u} \rangle \boldsymbol{v} - \boldsymbol{u}$。此鏡射操作為對 $\boldsymbol{u}$ 的線性變換。

§3.2 Grover 演算法

記得 $\mathcal{G} = \mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}\mathrm{U}_{f,\pm}$ 且當 $x \in \{0,1\}$ 時 $\mathrm{U}_{f,\pm}|x\rangle = (-1)^{f(x)}|x\rangle$。

1. $\mathrm{U}_{f,\pm}$ 的操作即是對 $|B\rangle$ 做鏡射：因為 $\langle G|B\rangle = 0$ 且
$$\mathrm{U}_{f,\pm}\big(a|G\rangle + b|B\rangle\big) = -a|G\rangle + b|B\rangle\,.$$
2. $\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}$ 的操作即是對均勻疊加態 $|U\rangle$ 做鏡射：首先注意到對量子態 $|\psi\rangle$ 做鏡射這個線性變換可表示成
$$2|\psi\rangle\langle\psi| - \mathrm{I}$$
因為對一個量子態 $|\phi\rangle$ 我們有
$$(2|\psi\rangle\langle\psi| - \mathrm{I})|\phi\rangle \equiv 2\langle\psi|\phi\rangle|\psi\rangle - |\phi\rangle\,.$$
因此 $\mathrm{R} = 2|0\rangle^{\otimes n}\langle 0|^{\otimes n} - \mathrm{I}$ 因而我們有
$$\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n} = \mathrm{H}^{\otimes n}(2|0\rangle^{\otimes n}\langle 0|^{\otimes n} - \mathrm{I})\mathrm{H}^{\otimes n} = 2|U\rangle\langle U| - \mathrm{I}\,.$$

§3.2 Grover 演算法

記得 $\mathcal{G} = \mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}\mathrm{U}_{f,\pm}$ 且當 $x \in \{0,1\}$ 時 $\mathrm{U}_{f,\pm}|x\rangle = (-1)^{f(x)}|x\rangle$。

1. $\mathrm{U}_{f,\pm}$ 的操作即是對 $|B\rangle$ 做鏡射：因為 $\langle G|B\rangle = 0$ 且
$$\mathrm{U}_{f,\pm}\big(a|G\rangle + b|B\rangle\big) = -a|G\rangle + b|B\rangle\,.$$
2. $\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}$ 的操作即是對均勻疊加態 $|U\rangle$ 做鏡射：首先注意到對量子態 $|\psi\rangle$ 做鏡射這個線性變換可表示成
$$2|\psi\rangle\langle\psi| - \mathrm{I}$$
因為對一個量子態 $|\phi\rangle$ 我們有
$$(2|\psi\rangle\langle\psi| - \mathrm{I})|\phi\rangle \equiv 2\langle\psi|\phi\rangle|\psi\rangle - |\phi\rangle\,.$$
因此 $\mathrm{R} = 2|0\rangle^{\otimes n}\langle 0|^{\otimes n} - \mathrm{I}$ 因而我們有
$$\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n} = \mathrm{H}^{\otimes n}(2|0\rangle^{\otimes n}\langle 0|^{\otimes n} - \mathrm{I})\mathrm{H}^{\otimes n} = 2|U\rangle\langle U| - \mathrm{I}\,.$$

§3.2 Grover 演算法

記得 $\mathcal{G} = \mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}\mathrm{U}_{f,\pm}$ 且當 $x \in \{0,1\}$ 時 $\mathrm{U}_{f,\pm}|x\rangle = (-1)^{f(x)}|x\rangle$。

1. $\mathrm{U}_{f,\pm}$ 的操作即是對 $|B\rangle$ 做鏡射：因為 $\langle G|B\rangle = 0$ 且
$$\mathrm{U}_{f,\pm}\big(a|G\rangle + b|B\rangle\big) = -a|G\rangle + b|B\rangle\,.$$
2. $\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}$ 的操作即是對均勻疊加態 $|U\rangle$ 做鏡射：首先注意到對量子態 $|\psi\rangle$ 做鏡射這個線性變換可表示成
$$2|\psi\rangle\langle\psi| - \mathrm{I}$$
因為對一個量子態 $|\phi\rangle$ 我們有
$$(2|\psi\rangle\langle\psi| - \mathrm{I})|\phi\rangle \equiv 2\langle\psi|\phi\rangle|\psi\rangle - |\phi\rangle\,.$$
因此 $\mathrm{R} = 2|0\rangle^{\otimes n}\langle 0|^{\otimes n} - \mathrm{I}$ 因而我們有
$$\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n} = \mathrm{H}^{\otimes n}(2|0\rangle^{\otimes n}\langle 0|^{\otimes n} - \mathrm{I})\mathrm{H}^{\otimes n} = 2|U\rangle\langle U| - \mathrm{I}\,.$$

§3.2 Grover 演算法

記得 $\mathcal{G} = \mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}\mathrm{U}_{f,\pm}$ 且當 $x \in \{0,1\}$ 時 $\mathrm{U}_{f,\pm}|x\rangle = (-1)^{f(x)}|x\rangle$。

1. $\mathrm{U}_{f,\pm}$ 的操作即是對 $|B\rangle$ 做鏡射：因為 $\langle G|B\rangle = 0$ 且
$$\mathrm{U}_{f,\pm}\big(a|G\rangle + b|B\rangle\big) = -a|G\rangle + b|B\rangle\,.$$
2. $\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}$ 的操作即是對均勻疊加態 $|U\rangle$ 做鏡射：首先注意到對量子態 $|\psi\rangle$ 做鏡射這個線性變換可表示成
$$2|\psi\rangle\langle\psi| - \mathrm{I}$$
因為對一個量子態 $|\phi\rangle$ 我們有
$$(2|\psi\rangle\langle\psi| - \mathrm{I})|\phi\rangle \equiv 2\langle\psi|\phi\rangle|\psi\rangle - |\phi\rangle\,.$$
因此 $\mathrm{R} = 2|0\rangle^{\otimes n}\langle 0|^{\otimes n} - \mathrm{I}$ 因而我們有
$$\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n} = \mathrm{H}^{\otimes n}(2|0\rangle^{\otimes n}\langle 0|^{\otimes n} - \mathrm{I})\mathrm{H}^{\otimes n} = 2|U\rangle\langle U| - \mathrm{I}\,.$$

§3.2 Grover 演算法

假設我們知道滿足搜尋條件的資料之占比 $\varepsilon = t/N$，則 Grover 演算法可重述如下：

1. 設定均勻疊加態 $|U\rangle = \mathrm{H}^{\otimes n}|0\rangle^{\otimes n}$。
2. 以順序重複執行 $k = \mathcal{O}(1/\sqrt{\varepsilon})$ 次下列的操作：
   - ⓐ 對 $|B\rangle$ 做鏡射（即執行 $\mathrm{U}_{f,\pm}$ 這個操作）；
   - ⓑ 對均勻疊加態 $|U\rangle$ 做鏡射（即執行 $\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}$ 這個操作）。
3. 測量所有的 qubits 並查看所得是否滿足搜尋條件。

§3.2 Grover 演算法

有一個非常簡單的幾何看法可用來解釋 Grover 演算法為何成立。我們由量子態 $|U\rangle = \sin\theta|G\rangle + \cos\theta|B\rangle$ 開始，首次的鏡射操作ⓐ 與ⓑ 將量子態與 $|B\rangle$ 的夾角由輸入時的 θ 增加到輸出時的 3θ，因此執行了一次 Grover iterate 將疊加態 $|U\rangle$ 往 good state $|G\rangle$ 移動，如圖 10 所示：

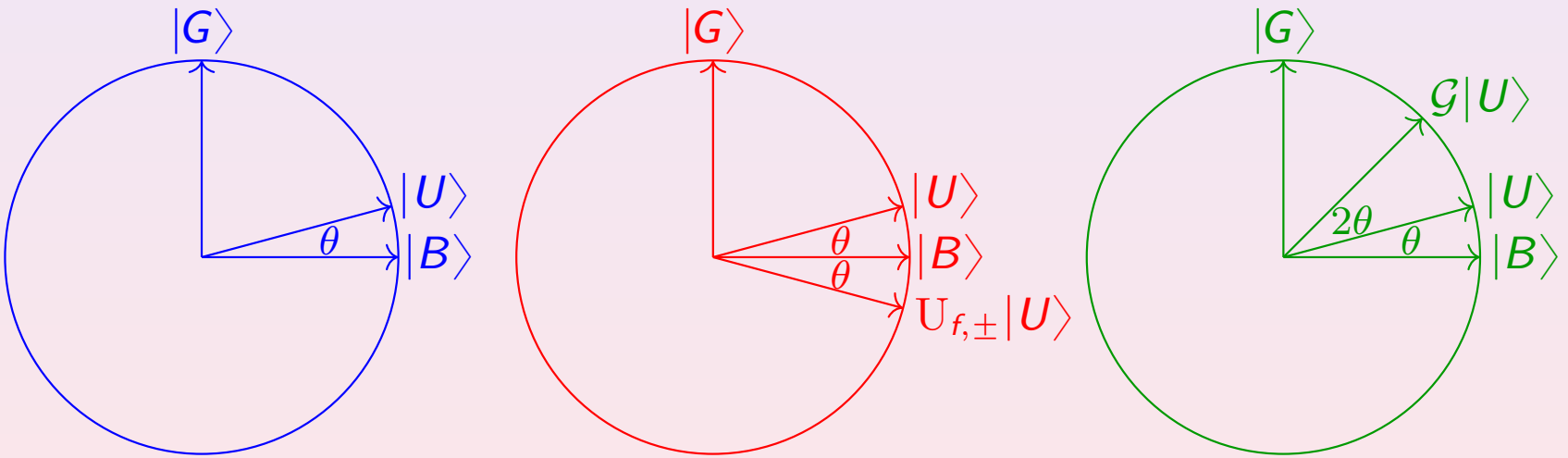


Figure 10: 首次 Grover iterate：（左）從 $|U\rangle$ 開始，（中）對 $|B\rangle$ 做鏡射得到 $\mathrm{U}_{f,\pm}|U\rangle$，（右）對 $|U\rangle$ 做鏡射得到 $\mathcal{G}|U\rangle$

§3.2 Grover 演算法

每一次的鏡射操作 ⓐ 與 ⓑ 都將輸出與 $|B\rangle$ 的夾角增加 2θ。因此，經過 k 次 ⓐ 與 ⓑ 的鏡射操作後，最終的量子態變成

$$\sin((2k+1)\theta)|G\rangle + \cos((2k+1)\theta)|B\rangle.$$

接下來對此最終量子態進行測量，得到滿足搜尋條件的資料的機率為 $P_k = \sin^2((2k+1)\theta)$。我們希望 P_k 愈接近 1 愈好，所以我們傾向於選擇滿足 $(2k+1)\theta = \frac{\pi}{2}$ 這樣的 k（其對應到的 $P_k = \sin^2\frac{\pi}{2} = 1$）。例如若 $t = \frac{N}{4}$，則 $\theta = \frac{\pi}{6}$ 那麼對應的 $k = 1$（亦即只進行一個 Grover iterate 即可）。不過不幸的是一般的滿足 $(2k+1)\theta = \frac{\pi}{2}$ 的 k 並非整數，而我們只能進行整數次的 Grover iterate。

§3.2 Grover 演算法

每一次的鏡射操作 ⓐ 與 ⓑ 都將輸出與 $|B\rangle$ 的夾角增加 2θ。因此，經過 k 次 ⓐ 與 ⓑ 的鏡射操作後，最終的量子態變成

$$\sin((2k+1)\theta)|G\rangle + \cos((2k+1)\theta)|B\rangle.$$

接下來對此最終量子態進行測量，得到滿足搜尋條件的資料的機率為 $P_k = \sin^2((2k+1)\theta)$。我們希望 P_k 愈接近 1 愈好，所以我們傾向於選擇滿足 $(2k+1)\theta = \frac{\pi}{2}$ 這樣的 k（其對應到的 $P_k = \sin^2\frac{\pi}{2} = 1$）。例如若 $t = \frac{N}{4}$，則 $\theta = \frac{\pi}{6}$ 那麼對應的 $k = 1$（亦即只進行一個 Grover iterate 即可）。不過不幸的是一般的滿足 $(2k+1)\theta = \frac{\pi}{2}$ 的 k 並非整數，而我們只能進行整數次的 Grover iterate。

§3.2 Grover 演算法

雖然我們通常不能挑到一個整數 k 讓進行 k 次 Grover iterate 的 Grover 演算法在測量後就吐出滿足搜尋條件的資料，我們仍可以挑一個最靠近 $\tilde{k} = \frac{\pi}{4\theta} - \frac{1}{2}$ 的整數 k 來進行 Grover 演算法。這樣的 k 所對應的 P_k 會很靠近 1 - 意即失敗率很低（在 $t \ll N$ 的假設之下）：

$$\begin{aligned} 1 - P_k &= \cos^2((2k+1)\theta) = \cos^2((2\tilde{k}+1)\theta + 2(k-\tilde{k})\theta) \\ &= \cos^2\left(\frac{\pi}{2} + 2(k-\tilde{k})\theta\right) \\ &= \sin^2(2(k-\tilde{k})\theta) \leqslant \sin^2\theta = \frac{t}{N}, \end{aligned}$$

在此我們用了 $|k - \tilde{k}| \leqslant 1/2$ 這個最接近 $\tilde{k}$ 的整數 k 會滿足的性質。因為 $\arcsin\theta \geqslant \theta$，所需要 query 次數 $k \leqslant \frac{\pi}{4\theta} \leqslant \frac{\pi}{4}\sqrt{\frac{N}{t}}$。

§3.2 Grover 演算法

雖然我們通常不能挑到一個整數 k 讓進行 k 次 Grover iterate 的 Grover 演算法在測量後就吐出滿足搜尋條件的資料，我們仍可以挑一個最靠近 $\tilde{k} = \frac{\pi}{4\theta} - \frac{1}{2}$ 的整數 k 來進行 Grover 演算法。這樣的 k 所對應的 P_k 會很靠近 1 - 意即失敗率很低（在 $t \ll N$ 的假設之下）：

$$\begin{aligned} 1 - P_k &= \cos^2((2k+1)\theta) = \cos^2((2\tilde{k}+1)\theta + 2(k-\tilde{k})\theta) \\ &= \cos^2\left(\frac{\pi}{2} + 2(k-\tilde{k})\theta\right) \\ &= \sin^2(2(k-\tilde{k})\theta) \leqslant \sin^2\theta = \frac{t}{N}, \end{aligned}$$

在此我們用了 $|k - \tilde{k}| \leqslant 1/2$ 這個最接近 $\tilde{k}$ 的整數 k 會滿足的性質。因為 $\arcsin\theta \geqslant \theta$，所需要 query 次數 $k \leqslant \frac{\pi}{4\theta} \leqslant \frac{\pi}{4}\sqrt{\frac{N}{t}}$。

§3.2 Grover 演算法

Example

考慮搜尋滿足 $f(x_1x_2x_3) = x_1 \cdot (1 - x_2) \cdot x_3$ 這個由 $\{0,1\}^3$ 映至 $\{0,1\}$ 的函數 f 其函數值為 1 的資料。首先，我們先開啟上帝視角得知 f 只把 $(101)_2$ 送到 1，因此 U_f 的作用等同於前三個 qubits（即控制位元）分別為 1、0、1 時「否定」目標位元（意即在最後一個位元加上 **NOT** 閘），也因此 U_f 可用下面的量子電路實現

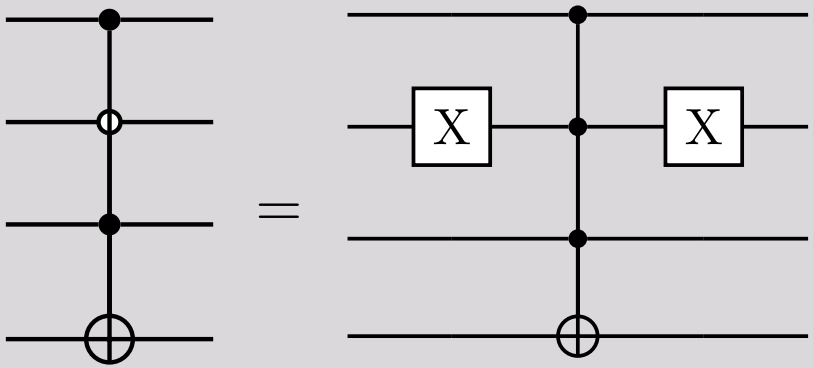


Figure 11: 本例中給定之 f 對應的 U_f 之量子電路

§3.2 Grover 演算法

Example (con't)

我們也要製造 R 的量子電路。事實上若定義

$$g(x_1x_2x_3) = \begin{cases} 0 & \text{if } (x_1x_2x_3)_2 = 0\,, \\ 1 & \text{otherwise}\,, \end{cases}$$

則由 $\mathrm{U}_{g,\pm}|x\rangle = (-1)^{g(x)}|x\rangle$ 很容易看出 $\mathrm{R} = \mathrm{U}_{g,\pm}$。而如同之前製造 U_g 的想法，U_g 的量子電路也很容易製造：

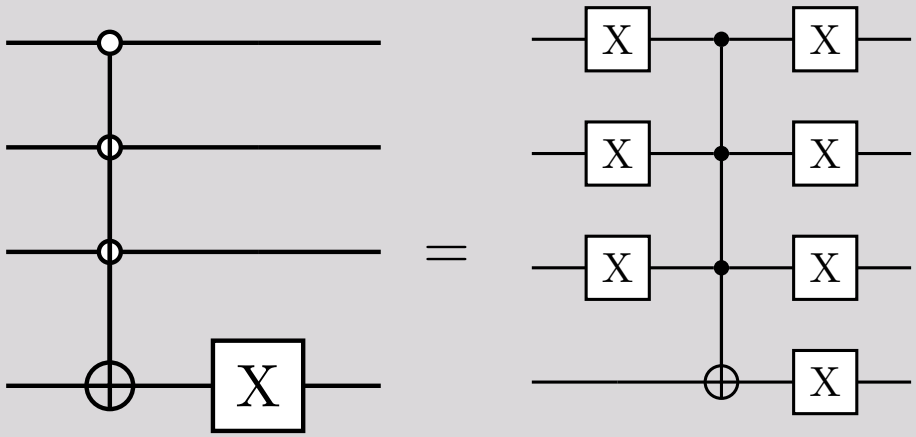


Figure 12: 本例中 U_g 之量子電路

§3.2 Grover 演算法

Example (con't)

因此，一個 Grover iterate（這是一個作用在 3 量子比特上的邏輯匣）加上第四個 qubit 的輸入是 $|-\rangle$ 其整組系統可用以下量子電路實現：

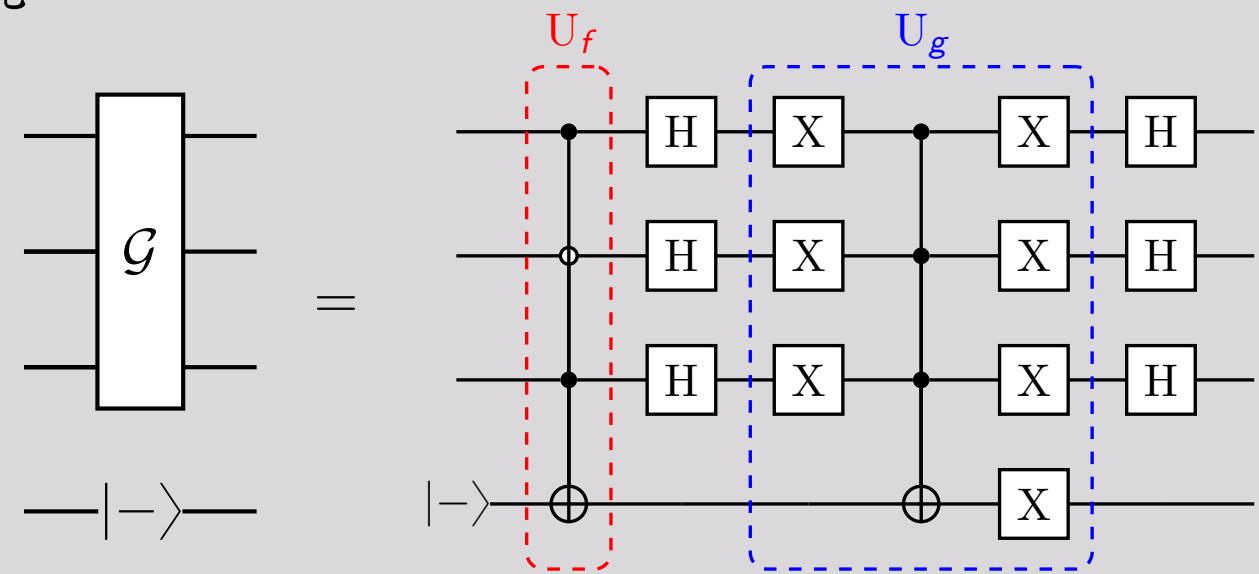


Figure 13: 本例中 "一個 Grover iterate" 的量子電路

有了一個 Grover iterate 的量子電路，整個 Grover 演算法的電路就可以輕易造出。

§3.2 Grover 演算法

Example (con't)

有沒有純 3-qubit 的量子電路？也就是說，我們能不能不引進第四個 qubit（並在這個 qubit 輸入 $|-\rangle$）來直接實現一個 Grover iterate 的量子電路？

答案是肯定的，不過在進行之前就必須講到在 3-qubit 量子邏輯閘中前兩個位元為控制位元而第三個位元為目標位元時的 multi-controlled 閘。我們將專注在以下四種 multi-controlled 閘：

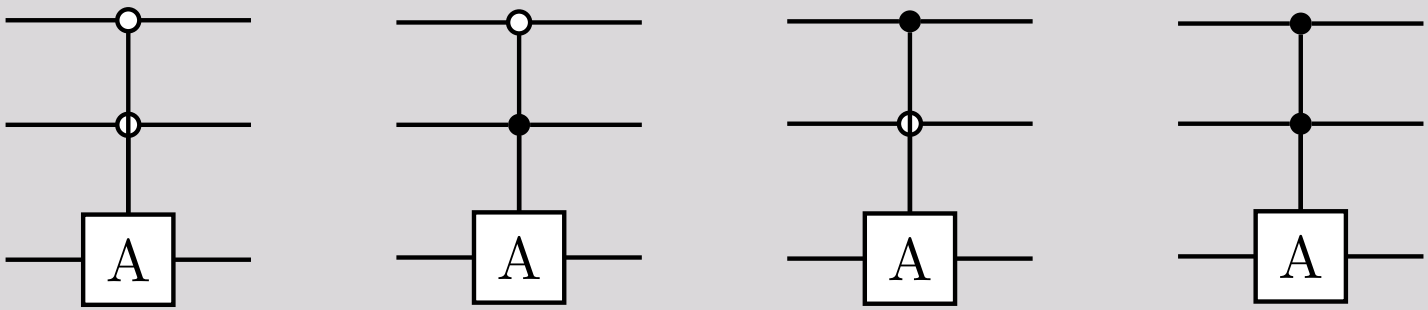


Figure 14: 四個特別的 multi-controlled 閘

在此 A 代表某個 1-qubit 閘。

§3.2 Grover 演算法

Example (con't)

若量子邏輯閘 A 的矩陣表示為 $[\mathrm{A}]$，然後 $\mathbf{I}$ 是 2×2 的單位矩陣，那麼上頁的四個 multi-controlled 閘的矩陣表示分別為

$$\begin{bmatrix} [\mathrm{A}] & & & \\ & \mathbf{I} & & \\ & & \mathbf{I} & \\ & & & \mathbf{I} \end{bmatrix}, \begin{bmatrix} \mathbf{I} & & & \\ & [\mathrm{A}] & & \\ & & \mathbf{I} & \\ & & & \mathbf{I} \end{bmatrix}, \begin{bmatrix} \mathbf{I} & & & \\ & \mathbf{I} & & \\ & & [\mathrm{A}] & \\ & & & \mathbf{I} \end{bmatrix}, \begin{bmatrix} \mathbf{I} & & & \\ & \mathbf{I} & & \\ & & \mathbf{I} & \\ & & & [\mathrm{A}] \end{bmatrix}.$$

注意到 $\mathrm{U}_{f,\pm}$ 只改變 $|5\rangle$ 的機率振幅而 R 改變了所以非零的機率振幅，因此其矩陣表示分別為

$$[\mathrm{U}_{f,\pm}] = \begin{bmatrix} \mathbf{I} & & & \\ & \mathbf{I} & & \\ & & \begin{bmatrix} -1 & \\ & 1 \end{bmatrix} & \\ & & & \mathbf{I} \end{bmatrix} \quad \text{與} \quad [\mathrm{R}] = \begin{bmatrix} \begin{bmatrix} 1 & \\ & -1 \end{bmatrix} & & & \\ & -\mathbf{I} & & \\ & & -\mathbf{I} & \\ & & & -\mathbf{I} \end{bmatrix}.$$

§3.2 Grover 演算法

Example (con't)

引入 Z 閘，其矩陣表示為 $[\mathrm{Z}] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$。注意到

$$[\mathrm{X}][\mathrm{Z}][\mathrm{X}] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} -1 & \\ & 1 \end{bmatrix},$$

因此 $\mathrm{U}_{f,\pm}$ 可由下面的量子電路實現：

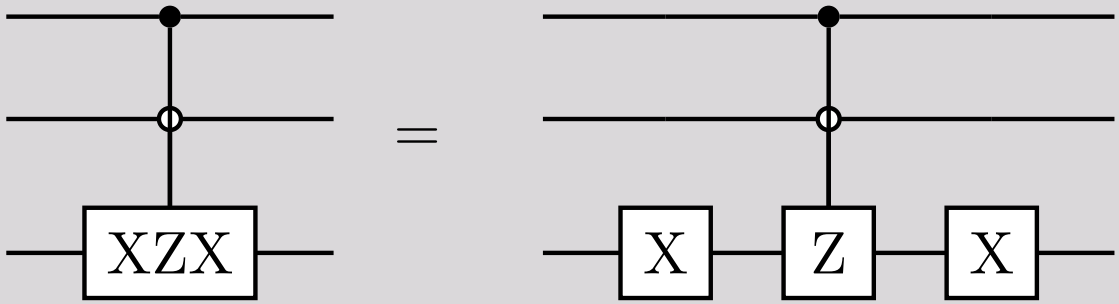


Figure 15: 本例中 $\mathrm{U}_{f,\pm}$ 的量子電路

§3.2 Grover 演算法

Example (con't)

另外注意到若 Grover iterate 中的 R 改為 $-\mathrm{R}$（意即只將 $|0\rangle^{\otimes n}$ 的機率振幅乘上 -1 其餘不變），依此 Grover iterate 建立的演算法依然是有效的 Grover 演算法（因為只改變了 global phase）。而

$$[-\mathrm{R}] = \begin{bmatrix} \begin{bmatrix} -1 & \\ & 1 \end{bmatrix} & & \\ & \mathrm{I} & \\ & & \mathrm{I} \end{bmatrix}$$

可用下面的量子電路實現：

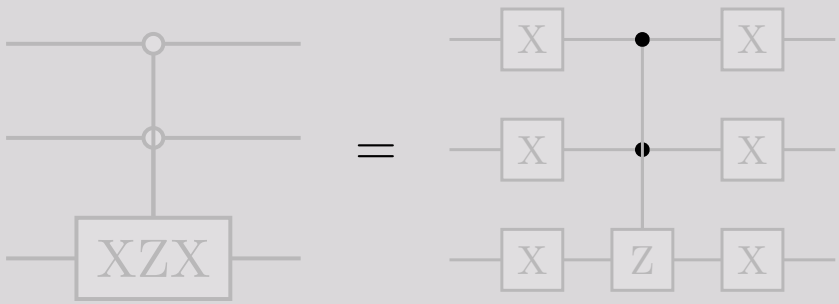


Figure 16: 本例中 "R" 的量子電路

§3.2 Grover 演算法

Example (con't)

另外注意到若 Grover iterate 中的 R 改為 $-\mathrm{R}$（意即只將 $|0\rangle^{\otimes n}$ 的機率振幅乘上 -1 其餘不變），依此 Grover iterate 建立的演算法依然是有效的 Grover 演算法（因為只改變了 global phase）。而

$$[-\mathrm{R}] = \begin{bmatrix} \begin{bmatrix} -1 & \\ & 1 \end{bmatrix} & & & \\ & \mathbf{I} & & \\ & & \mathbf{I} & \\ & & & \mathbf{I} \end{bmatrix}$$

可用下面的量子電路實現：

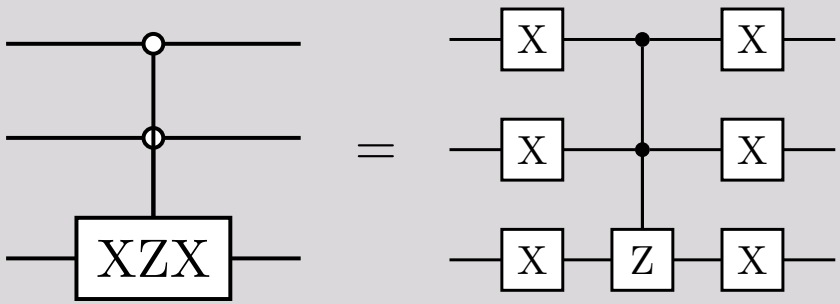


Figure 16: 本例中 " R" 的量子電路

§3.2 Grover 演算法

Example (con't)

綜合以上，我們得到一個 Grover iterate 的量子電路：

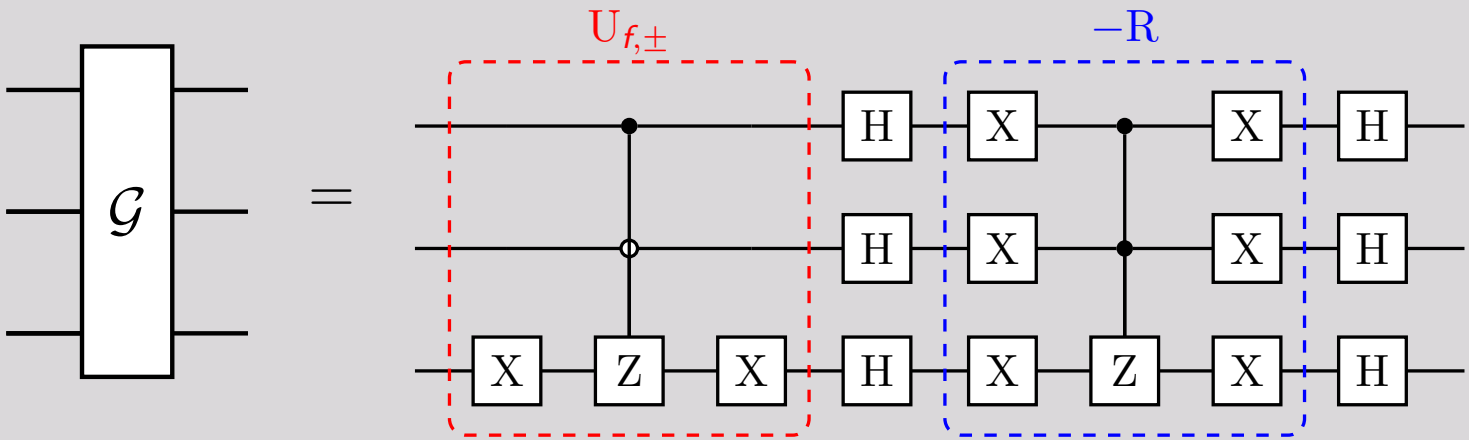


Figure 17: 本例中 "一個 Grover iterate" 的量子電路

有了一個 Grover iterate 的量子電路，整個 Grover 演算法的電路就可以輕易造出。

§3.2 Grover 演算法

Example (con't)

綜合以上，我們得到一個 Grover iterate 的量子電路：

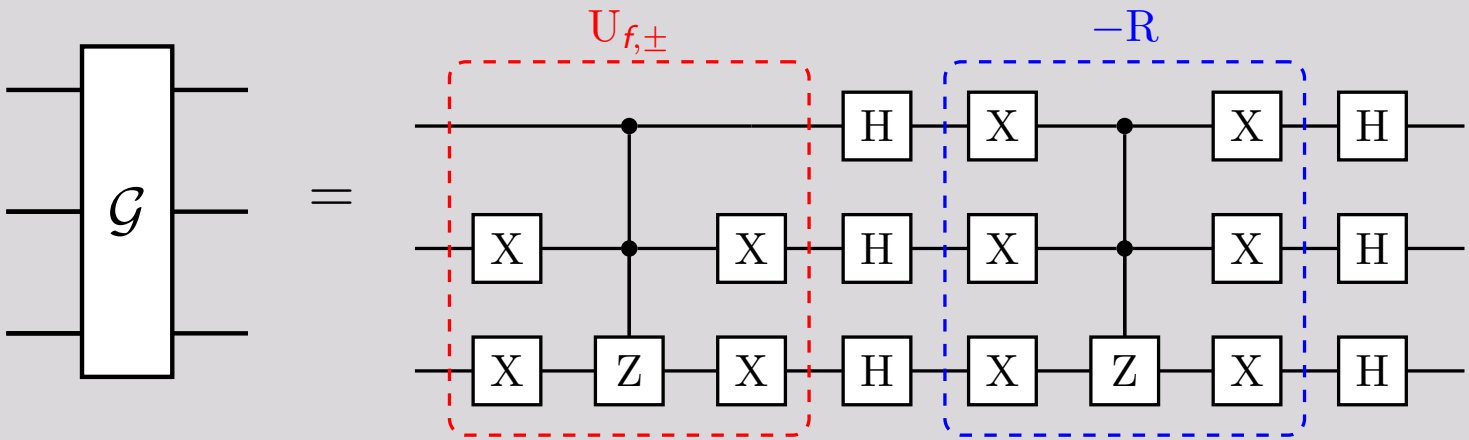


Figure 17: 本例中 "一個 Grover iterate" 的量子電路

有了一個 Grover iterate 的量子電路，整個 Grover 演算法的電路就可以輕易造出。