自然數面等機率量子骰探討與研究

11934　Infor 33rd 蕭登鴻

壹、 研究動機

學習基本量子運算的過程中，我們學到可以利用疊加態的特性，讓單一或是複數個量子位元出現不同的可能性組合。

利用此特性，將其作為骰子來看的話，如果將所有的結果都設為等機率出現，就可做出絕對公正且公平的骰子。此骰子不同於電腦的亂數，不會存在偽亂數的可能性。

我們在做出 $2^n, n \in \mathbb{N}$ 面骰之後，又會想做出其餘的奇、偶面骰。但在持續做下去後，我們便會因為迴路複雜度漸增與數字的增大，開始思考是否有通式，或是有系統的方式做出量子骰。但在此時，我們又會反問自己 – 自然數面等機率骰是否存在？

貳、 定義

此文章中探討的量子骰為嚴謹定義的 N 面等機率量子骰，指的是利用量子位元可以處於疊加態的特性，將單一或是數個量子位元通過各種量子閘後，讓單一或是數個量子位元在最終進行測量時結果可能性只有 $|0_2\rangle \sim |N-1_2\rangle$ 共 N 種結果，且每種結果測量到的機率都相同。

參、 運算與證明

經過定義，可得出將自然數個量子位元通過量子閘之後的應有量子態為：

$$|\varphi\rangle = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}|n_2\rangle$$

接下來，必須要先知道 N 種結果至少會需要多少個量子位元來表示。因 α 個量子位元至多可表示 2^α 種結果，所以表達 N 種結果所需最少量子位元數為：

$$i_N = \lceil log_2 N \rceil$$

另外，運用 2 進位的想法，N 可以被拆成數個 2 的 n_k 次方。同樣的，N 面骰也可以被拆成數個 2^{n_k} 面骰來實作。而作出 2^n 面骰所施加

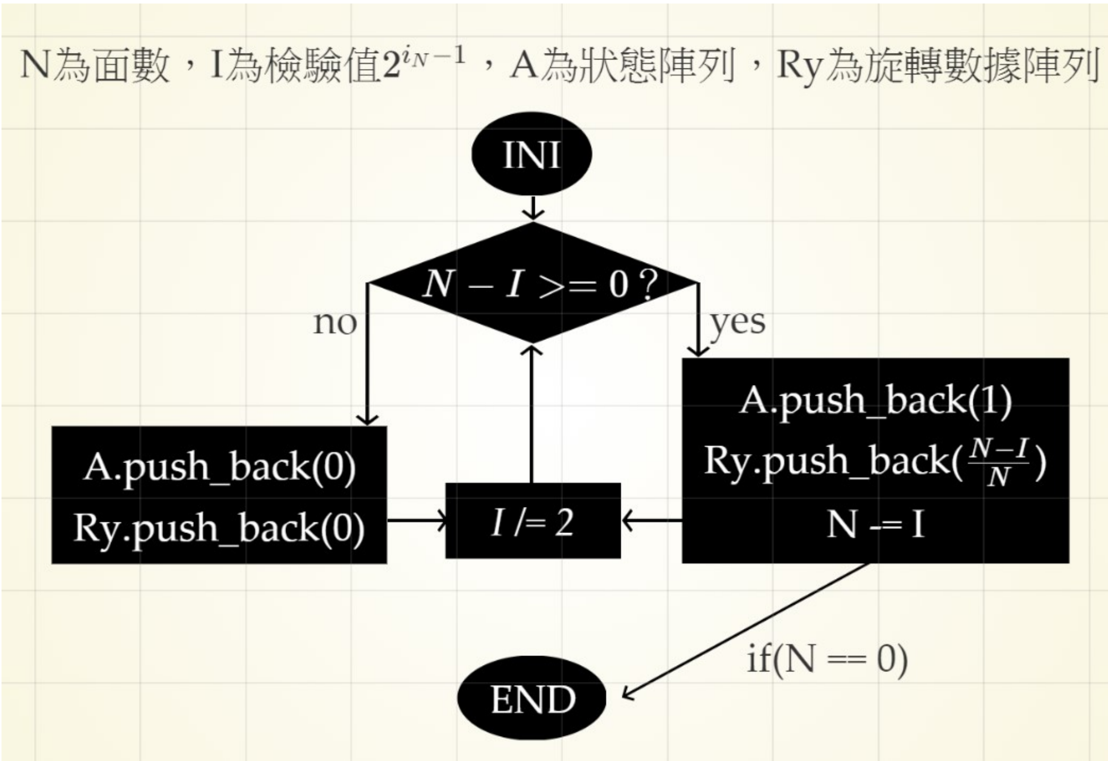
的 $H\,Gate$ 數為 $log_2\,2^n$。由此邏輯可得知：

$$\forall N \in \mathbb{N}, \exists!\, N_2$$

也就是，只要是屬於自然數面的等機率量子骰，都可以被拆為多個 2^n 面等機率量子骰，因此可得知自然數面等機率量子骰的存在。

肆、 實作運算方式

製作迴路時，首先要注意迴路的量子位元編號，是由上到下從 0 開始排序；另外，要從被分解的 2^{n_k} 中以大至小做。我們可以從以下的算法得出要對每個量子位元進行的操作。



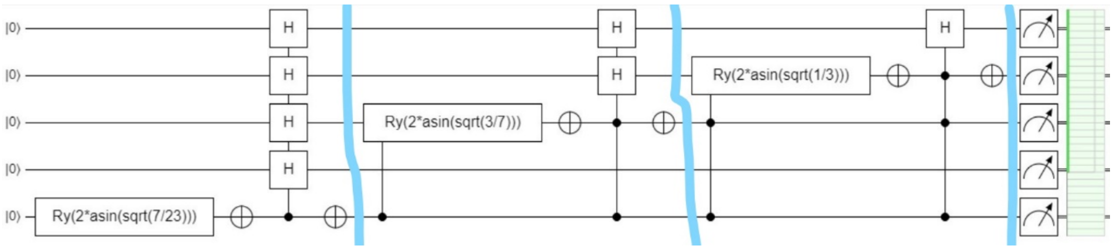
其意義為：將 N 持續減去 2^{n_k}，n_k 由 $i-1$ 至 0。如果大於 0 代表 N 的分解中包含 2^{n_k}；如果小於 0 代表沒有；如果等於 0，代表 N 已被分解完，所以就此停止。

另外，如果總共有 M 種狀態要表示，k 個量子位元已表示完 2^k 個，那代表剩下的 $M-2^k$ 種狀態要在第 $k+1$ 個量子位元為進位的狀態下表示。

因此，其中的 Ry 為儲存第 $k+1$ 個量子位元為 $|1\rangle$ 的機率，意義為：N 種狀態中的 $N-I$ 種狀態是在第 $k+1$ 個量子位元為 $|0\rangle$（未進位）時表示；剩餘的 $N-I$ 種狀態是在第 $k+1$ 個量子位元為 $|1\rangle$（已進位）時表示；因此使用 $Ry\,Gate$ 對第 $k+1$ 個量子位元的量子態進行

改變，使其機率符合上述進位狀態，並利用量子邏輯閘判斷在第 $k+1$ 個量子位元為 $|0\rangle$ 時將後方 k 個量子位元施加 $H\,Gate$ 變為 2^k 個等機率結果。

如果使用此運算方式，便可快速地做出 N 面骰（以 23 為例）：



1. $23-16>0\;(=7)$
2. $7-8<0\;(=-1)$
3. $7-4>0\;(=3)$
4. $3-2>0\;(=1)$
5. $1-1=0$

$\mathrm{A}=[1,\ 0,\ 1,\ 1,\ 1],\mathrm{Ry}=\left[\frac{7}{23},\ 0,\ \frac{3}{7},\ \frac{1}{3}\right]$

$\rightarrow 23=16+4+2+1$

- 第一藍色線段前

  因 23 種狀態中有 7 種狀態需要進位，所以利用 $Ry\left(\sqrt{\frac{7}{23}}\right)$ 改變 Q_5 的機率，並利用 cH 判斷**如果 $\boldsymbol{Q_5}$ 為 $|\mathbf{0}\rangle$（未進位）**，就將 $Q_{4\sim0}$ 施加 $H\,Gate$。

- 第二藍色線段前

  剩下的 7 種狀態有 3 種需進位，所以利用 $Ry\left(\sqrt{\frac{3}{7}}\right)$ 改變 Q_2 的狀態，並且判斷如果 **$\boldsymbol{Q_5}$ 為 $|\mathbf{1}\rangle$（已進位）且 $\boldsymbol{Q_2}$ 為 $|\mathbf{0}\rangle$（未進位）**，就將 $Q_{1\sim0}$ 施加 $H\,Gate$。

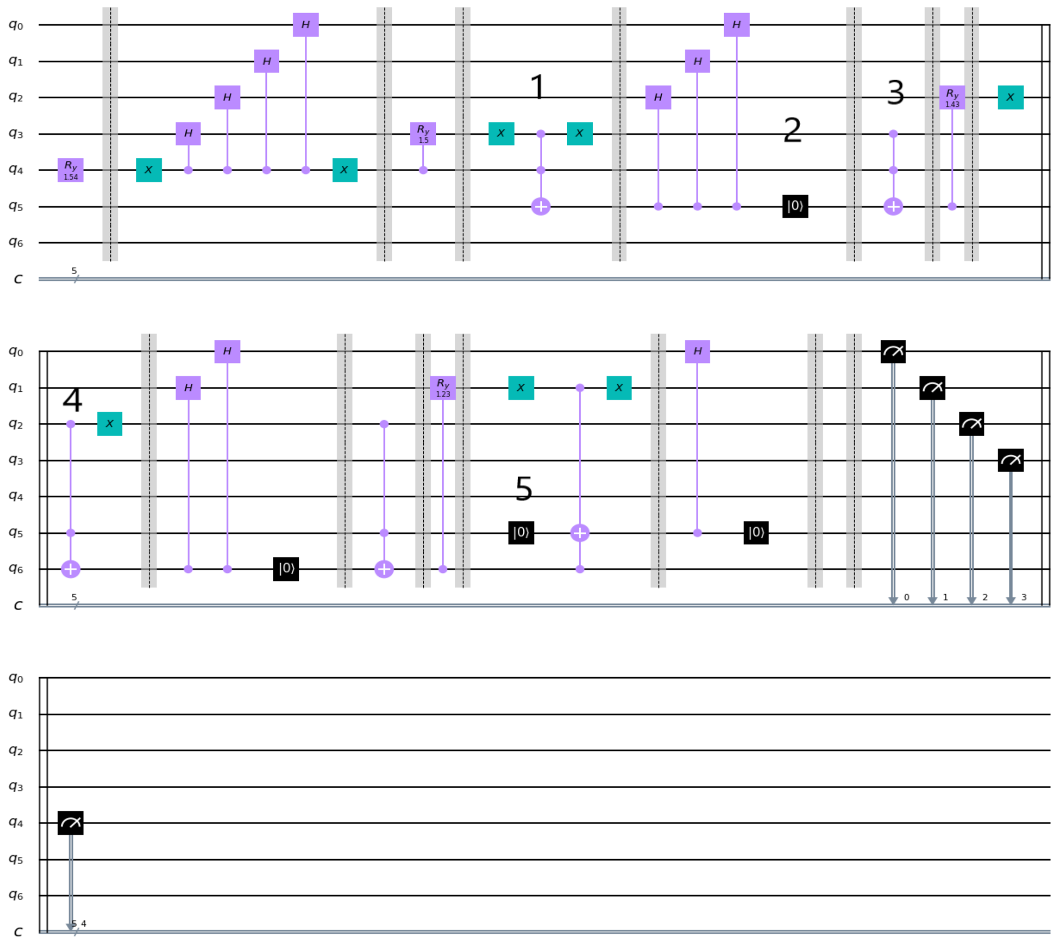
- 第三藍色線段前

  剩下的 3 種狀態有 1 種需進位，所以利用 $Ry\left(\sqrt{\frac{1}{3}}\right)$ 改變 Q_1 的狀態，並且判斷如果 **$\boldsymbol{Q_5}$ 為 $|\mathbf{1}\rangle$（已進位）且 $\boldsymbol{Q_2}$ 為 $|\mathbf{1}\rangle$（已進位）且 $\boldsymbol{Q_1}$ 為 $|\mathbf{0}\rangle$（未進位）**，就將 Q_0 施加 $H\,Gate$；剩下的 1 種狀態不需要施加 cH 即可完成（$log_2 1=0$）。

伍、 程式實作序論

利用以上方法與邏輯，我們可以嘗試寫出一個可以印出嚴謹定義的 N 面等機率量子骰的迴路程式的程式，但是在此之前，我們還必須要解決量子邏輯閘的限制問題。

在接下來將使用的量子程式語言中，將不存在操控位元數大於 1 個 cH 與 cRy，因此必須要藉由 ccX 將先前的訊息與現在的訊息結合，儲存到另外用來儲存訊息（不在最終測量範圍中）的量子位元，同時利用 $reset$ 將不需要的量子位元中的資訊重置，以達到重複利用的效果。以下為實際的的作法（以 31 面骰為例）：



可從中看到，首先利用 ccX 將需判斷的資訊丟入儲存資訊的量子位元中 [1]，利用 cH 提取完資訊後再利用 $reset$ 將已使用過且不再需要的資訊刪除後 [2]，再將新的資訊儲存進去，作為下一階段的判斷根據 [3]。

而到下一階段時，又將資訊儲存在另一個量子位元中 [4]，重複上述的操作；但是再下一個階段時，就必須要先使用 $reset$ 清空上一個使用過的量子位元中的東西 [5]，才可再次利用其量子位元。

陸、 程式實作

在程式中，我們將會利用在第四點的演算法，先將 N 分解；並將狀態儲存在 $std::vector < int >$ 中、將機率儲存在 $std::vector < std::string >$ 中。

拆解 N 後，要先判斷有幾個 H 將會是恆必須要施加於迴路上（代表 N 的質因數中有幾個 2），而其數量將會等於 $T - A.size(), T = i_N$，並將此數量儲存至名為 SH $(Stable\_H\_Count)$ 的變數。

如此一來，就可以先判別所有的 cH 只要施加到 Q_{SH}，最後再將編號 $Q_{SH\sim 0}$ 統一施加 H 即可。

接下來用變數 i（從 0 開始至 $Ry.size() - 1$）代表層數（第 0 層為 Q_{i-1} 的迴路，第 1 層為 Q_{i-2} 的迴路），從 Q_{T-1-i} 開始至 Q_0 一層層檢視時，將利用 $AC(Active\ Count)$ 代表至此層有多少個有進位可能性的量子位元數、$LA(Last\ Active)$ 代表上一個儲存**前方量子位元都為 $|\mathbf{1}\rangle$** 此資訊的量子位元編號，因此可以得知在 $i > 0$（非第一層）的狀況下，要儲存訊息的量子位元編號將會是 $T + (AC \% 2)$。

而這些都是為了要將儲存資訊的 ccX 中的量子位元編號公式化，公式為：

$$ccX\big(T - 1 - i, LA, T + (AC \% 2)\big)$$

在交替使用儲存資訊的量子位元時，我們必須要注意一個量子位元在使用前是否是乾淨的，因此在 $AC >= 4$ 的狀況下，必須要在 ccX 執行前先利用 $reset$ 清空要寫入資訊的量子位元。

而 $AC >= 4$ 此條件，是因為如果 $AC = 1$，將不會需要用到儲存資訊的量子位元；

如果 $AC = 2$，將會利用到一個儲存資訊的量子位元；

如果 $AC = 3$，將會需要用到 2 個儲存資訊的量子位元；

如果 $AC = 4$，將會重複用到第一個儲存資訊的量子位元，因此就需執行清理的動作。同時也會利用此邏輯，在輸出迴路之前先判斷此迴路總共會需要多少個量子位元（T +儲存資訊的量子位元數）。

以下為使用上述邏輯寫成的程式的網址：

https://slides.com/phantom0174/qd_p

柒、 未來展望

因為現在的量子電腦硬體技術尚未發達，在操作量子閘、測量等操作時都會有著錯誤的可能性，所以應該要將迴路的寬度與深度都盡量減少，達到減少錯誤率的效果。

即使目前的方法已經是改進過的方法（使用 $reset()$ 有效地減少了迴路的寬度），但是迴路仍然有著很可觀的深度；因此未來將考慮利用質因數分解的方式，將 N 先拆解為數個由小至大的質因數面骰相乘，再利用此方式去製作各質數面骰。

最理想的目標，就是在製作質數面骰的過程中使用迴路優化方式，也就是使用更少的量子閘達成跟原本方式同樣的結果（同樣的量子態），以上方式都可以有效地減少迴路的深度。

最終，也希望可以將視野放寬到其他非等機率的量子骰，去了解其功用（如 Quantum Random Walk），並抱持著與嚴謹定義等機率量子骰同樣的熱情去探討與研究。