

آلگوریتم های ساده کوانتومی برای محاسبه

وحیدکریمی پور- دانشکده فیزیک - دانشگاه صنعتی شریف

۸ دی ۱۳۹۱

۱ مقدمه

کامپیوتر کوانتومی چیست؟ چگونه می توان از مکانیک کوانتومی برای محاسبه استفاده کرد؟ آیا یک کامپیوتر کوانتومی قدرتی فراتر از کامپیوترهای کلاسیک دارد؟ اگر چنین است، کدام خصلت های مکانیک کوانتومی این توانایی را بوجود می آورند؟ هدف ما در این درس فراهم آوردن پاسخ هایی جزئی برای این سوال هاست. نخست به سوال اول می پردازیم. می دانیم که در کامپیوترهای کلاسیک هر نوع داده ای در بیت های کلاسیک ذخیره می شود. یک بیت کلاسیک صرف نظر از نوع سخت افزاری که برای آن به کار رفته است، یک سیستم فیزیکی است که می تواند در دو حالت ^۰ یا ^۱ قرار گیرد. از اصطلاح حافظه کلاسیک ^۱ برای مجموعه ای از بیت ها که می تواند داده های گسترده تری را در خود ذخیره کند. این که حافظه معمولاً به صورت دسته های هشت تایی به نام بایت یا بزرگتر از آن منظم شده است و یا این که نوع آن چگونه است (حافظه کوتاه مدت *Random Access Memory (RAM)* یا دائمی) فعلاً مورد توجه ما نیست. هر نوع پردازشی در نهایت به این معناست که ما بتوانیم توابع دلخواهی را برای داده هایی که در این حافظه ها ذخیره شده اند محاسبه کنیم. در درس گذشته دیدیم که چگونه می توان با استفاده از مدارهای منطقی کلاسیک، چه از نوع برگشت پذیر چه از نوع بازگشت ناپذیر آن، داده هایی را که در این حافظه ها ذخیره می شود پردازش کرد.

مفهوم مرکزی در کامپیوتر کوانتومی بیت کوانتومی یا کیوبیت ^۲ است. یک کیوبیت به یک سیستم کوانتومی گفته می شود که فضای هیلبرت

^۱ Classical Register

^۲ Qubit

آن دو بعدی است. این فضای هیلبرت را با C^2 نشان می دهیم.

$$C^2 := \left\{ \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \mid \alpha, \beta \in C \right\}. \quad (1)$$

معمول است که بردارهای پایه‌ی این فضا را با $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ و $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ نشان می دهند. یک کیوبیت بر خلاف بیت کلاسیک می تواند در ترکیبی از حالت های پایه فوق قرارگیرد. حالت یک کیوبیت با برداری در این فضای هیلبرت نشان داده می شود:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \quad (2)$$

یک حافظه کوانتومی n کیوبیتی^۳ عبارت است از مجموعه‌ای متشکل از n کیوبیت. فضای هیلبرت این حافظه عبارت است از $(C^2)^{\otimes n}$:

$$(C^2)^{\otimes n} = \left\{ \sum_{s_0, s_1, \dots, s_{n-1} \in \{0,1\}} \alpha_{s_0, s_1, \dots, s_{n-1}} |s_0, s_1, \dots, s_{n-1}\rangle \right\}, \quad (3)$$

و بعد آن برابر است با 2^n . دقت کنید که این حافظه می بایست چنان باشد که تمام بردارهای فضای هیلبرت آن قابل دسترسی باشند. بنابراین اگر دو کیوبیت را تنها کنارهم بگذاریم به این معنی نیست که یک حافظه دو کیوبیتی ساخته‌ایم. زیرا بردارهای فضای هیلبرت دو کیوبیت جداگانه به صورت زیر هستند:

$$|\phi\rangle \otimes |\phi'\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha'|0\rangle + \beta'|1\rangle), \quad (4)$$

یعنی به صورت ضرب تانسوری دو بردار از فضاهای هرکدام از کیوبیت ها نوشته می شوند و حال آنکه در فضای هیلبرت دو کیوبیت یعنی $(C^2)^{\otimes 2}$ بردارهایی وجود دارند که به صورت فوق قابل نوشتن نیستند، مثل بردار $|\phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. این نوع بردارها را اصطلاحاً بردارهای درهم تنیده می نامیم. برای آنکه یک بردار کلی به صورت

$$a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle \quad (5)$$

به صورت حاصل ضرب تانسوری دو بردار نوشته شود می بایست شرط $ad - bc = 0$ برقرار شود. بنابراین بردارهایی که به صورت ضرب تانسوری هستند مجموعه‌ای با اندازه صفر را در فضای هیلبرت دو کیوبیت تشکیل می دهند.

توازی کوانتومی: اولین تفاوت مهم کامپیوتر کوانتومی با کامپیوتر کلاسیک این است که یک حافظه کوانتومی می تواند در آن واحد در تمام حالت های بالقوه خود قرار بگیرد. این خصلت ناشی از برهم نهی حالت های کوانتومی است و اصطلاحاً توازی کوانتومی^۴ خوانده می شود. هرگاه

^۳n-quibt Quantum Register

^۴Quantum Parralellism

هر حالت $|s_0, s_1, \dots, s_{n-1}\rangle$ را برای کد کردن عدد دوتایی $s = (s_0, s_1, \dots, s_n)$ به کار ببریم حالتی مثل

$$|\phi\rangle = \sum_{s=0}^{2^n-1} \phi_s |s\rangle, \quad (6)$$

حالت کلی یک حافظه n بیتی است. بنابراین هرگاه حافظه را در حالت $|\psi\rangle$ قرار دهیم مثل این است که همزمان آن را در تمام حالت های $|s\rangle$ قرار داده ایم. البته واضح است که هرگاه حافظه را در پایه ی محاسباتی یعنی همان پایه ۰ ها و ۱ ها اندازه گیری کنیم تنها یکی از مقادیر s با احتمال $|\phi_s|^2$ بدست خواهند آمد. بنابراین توازی کوانتومی اگر چه یک خاصیت مهم حافظه است ولی این خاصیت را می بایست با ظرافت مورد بهره برداری قرار داد.

گیت کوانتومی: اگر اطلاعات را در کیوبیت ها یعنی حالت های کوانتومی ذخیره کنیم، به ناچار پردازش اطلاعات می بایست یا با عملگرهای یکانی که تحول را نشان می دهند و یا با اندازه گیری ها انجام بگیرند. معمولاً اصطلاح گیت کوانتومی برای عملگر یکانی بکار برده می شود. گیت کوانتومی هرگاه روی یک کیوبیت اثر کند آن را گیت تک کیوبیتی^۵ و هرگاه روی n تا کیوبیت اثر کند، گیت n کیوبیتی^۶ خوانده می شود. در درس های آینده راجع به گیت های چند کیوبیتی و طرز ساختن آنها به طور کلی به تفصیل بحث خواهیم کرد.

آلگوریتم کوانتومی: آلگوریتم کوانتومی در ساده ترین شکل آن به مجموعه ای از گیت های کوانتومی متوالی گفته می شود که روی یک حالت معین اولیه اثر می کنند و چنان تنظیم شده اند که حالت نهایی چنان باشد که پس از اندازه گیری های سنجیده ای روی آن جواب یک مسئله معین را با احتمال بسیار خوب در بر داشته باشد.

هدف ما در این درس این است که با چند آلگوریتم ساده کوانتومی آشنا شویم و ببینیم که چگونه کامپیوترهای کوانتومی و آلگوریتم های کوانتومی می توانند بعضی از مسائل را سریعتر از کامپیوترها و آلگوریتم های کلاسیک حل کنند.

۲ جعبه سیاه یا اوراکل

فرض کنید که جعبه سیاهی تابع f را به روی ورودی خود اعمال می کند. ما تنها می توانیم به این جعبه سیاه ورودی های مختلف بدهیم و خروجی های آن را ثبت کنیم. به این کار فراخوانی تابع می گویند. در اصطلاح علم کامپیوتر به چنین جعبه سیاهی یک اوراکل (*Oracle*) می گویند. در

^۵One qubit gate

^۶n-Qubit Gate

ساده ترین مثال فرض کنید که این تابع یک تابع یک بیتی است، یعنی $f: B_1 \rightarrow B_1$. ولی در حالت های کلی تر این تابع می تواند از n بیت به یک بیت یا از n بیت به m بیت باشد. اصطلاح جعبه سیاه را به این دلیل به کار می بریم که بسیاری از مسائل رایانش را می توان با استفاده از آن صورت بندی کرد. در این نوع مسائل هدف همواره این است که با دادن ورودی های مختلف به این جعبه سیاه یا تابع و مشاهده خروجی های آن بتوانیم خاصیتی از تابع را پیدا کنیم. در این نوع مسائل معمولاً نمی خواهیم نوع تابع را به طور کامل بفهمیم بلکه فقط یک خاصیت مشخص از تابع مورد نظر است. هم چنین در این نوع مسائل شرط های معینی را نیز در باره تابع یا جعبه سیاه دانسته شده فرض می کنیم. به این نوع شرط ها اصطلاحاً ”قول“ یا *promise* می گویند. در زیر چند نوع از این مسائل را طرح می کنیم و نشان می دهیم که الگوریتم های کوانتومی می توانند بهتر از الگوریتم های کلاسیک این مسائل را حل کنند. این مسائل به ترتیب از حالت خیلی ساده شروع شده و سپس به حالت های پیچیده تر ختم می شوند. نکته اصلی که در این الگوریتم و همه الگوریتم های کوانتومی وجود دارد استفاده هوشمندانه از خاصیت برهم نهی حالت ها است. هم چنین چیزی که در بسیاری از این الگوریتم ها مشترک است استفاده از این نکته است که جعبه سیاه به دلیل آنکه تابع را به صورت برگشت پذیر محاسبه می کند دارای دو ورودی و دو خروجی به ترتیب زیر است:

$$|x, y\rangle \rightarrow |x, y \oplus f(x)\rangle, \quad (7)$$

که در نتیجه آن

$$|x, -\rangle \equiv \frac{1}{\sqrt{2}}(|x, 0\rangle - |x, 1\rangle) \rightarrow \frac{1}{\sqrt{2}}(|x, f(x)\rangle - |x, f(x) + 1\rangle) \equiv (-1)^{f(x)}|x\rangle|-\rangle. \quad (8)$$

بنابراین وقتی که ورودی دوم را در حالت $|-\rangle$ نگه می داریم آنچه که در ورودی اول ظاهر می شود فازی است که توسط مقدار تابع $f(x)$ تعیین می شود بدون اینکه در ورودی دوم هیچ تغییری ایجاد شود. حال اگر ورودی اول را نیز در یک حالت برهم نهی یکنواخت تهیه کنیم آنگاه حالت خروجی نیز یک برهم نهی خواهد بود به این ترتیب که:

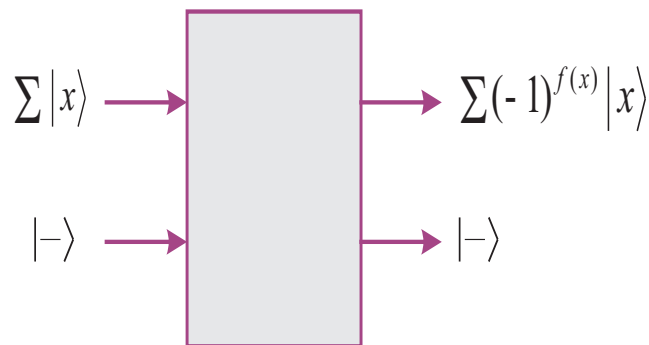
$$\sum_x |x, -\rangle \rightarrow \sum_x (-1)^{f(x)} |x\rangle |-\rangle. \quad (9)$$

توجه به این نکته کلید درک بسیاری از الگوریتم های است که در پی می آید. به طور خلاصه می توان این عمل را در دیاگرام زیر نشان داد: این نکته را یادآوری می کنیم که برای تهیه یک برهم نهی یکنواخت از تمامی حالت ها می بایست از عملگرهای هادامارد استفاده کنیم.

تمرین: نشان دهید که

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} (-1)^{x \cdot y} |y\rangle, \quad (10)$$

که در آن $N = 2^n$ است.



شکل ۱: یک نمونه از استفاده از توازی در الگوریتم های کوانتومی

۳ مسئله دویچ

یک تابع ساده از یک بیت به یک بیت را در نظر می گیریم: $f: B_1 \rightarrow B_1$. می خواهیم بینیم چند بار این تابع را فراخوانی کنیم، تا بتوانیم تابع f را تعیین کنیم. می دانیم که این تابع یکی از چهار تابعی است که در جدول ۳ نشان داده شده است. براحتی معلوم می شود برای تعیین تابع f می بایست دوبار تابع فراخوانی شود، یک بار با ورودی ۰ و بار دیگر با ورودی ۱ تا بتوان نوع تابع را تشخیص داد. می توانیم مسئله را ساده تر کنیم و بگوییم که دقیقاً نوع تابع را نمی خواهیم بلکه می خواهیم بفهمیم که آیا این تابع یک تابع ثابت است و یا یک تابع متوازن؟ منظور از تابع ثابت، تابعی است که خروجی اش همواره یک مقدار ثابت و مستقل از ورودی است (مثل تابع f_1 یا f_4) و منظور از تابع متوازن، تابعی است که خروجی اش به ازای نیمی از ورودی ها برابر با ۰ و به ازای نیمی دیگر برابر با ۱ است (مثل تابع f_2 و f_3). حال می پرسیم که با چند بار فراخوانی این تابع می توان به ثابت بودن یا متوازن بودن آن پی برد. بازهم کمی فکر کردن نشان می دهد که دو بار می بایست تابع فراخوانده شود.

با نگاهی به شکل refNewDeutsch می توان فهمید که اگر خروجی اول را اندازه بگیریم می توانیم براحتی ثابت بودن یا متوازن بودن تابع را تشخیص دهیم. این حالت را با $|\phi\rangle$ نشان می دهیم. در واقع اگر تابع ثابت باشد بدست می آوریم که منهای یک ضرب ثابت $|+\rangle = |\phi\rangle$ و اگر تابع متوازن باشد بدست می آوریم که $|\phi\rangle = |-\rangle$ که در آن $|+\rangle$ و $|-\rangle$ به ترتیب ویژه های مثبت و منفی عملگر σ_x هستند. در نتیجه اگر این خروجی را در پایه x اندازه بگیریم از روی مقدار +۱ یا -۱ که با قطعیت بدست می آوریم می توانیم تنها با یک بار فراخوانی جعبه سیاه خاصیت خواسته شده را بفهمیم.

f_4		f_3		f_2		f_1	
۱	۰	۱	۰	۰	۰	۰	۰
۱	۱	۰	۱	۱	۱	۰	۱

جدول ۱: انواع توابع یک بیتی

۴ الگوریتم کوانتومی برای مسئله دویچ - جوزا

مسئله دویچ جوزا (Deutsch – Josza) تعمیمی است از مسئله دویچ. فرض کنید که تابع $f: B_n \rightarrow B_1$ یک تابع n بیتی است و به ما گفته شده است که این تابع متوازن یا ثابت است یعنی مقادیر خروجی آن همه یا با هم مساوی هستند و یا نیمی از مقادیر خروجی اش صفر و نیمی صفر هستند. می دانیم که تعداد توابع ثابت برابر است با 2^{n-1} . اما تعداد توابع متوازن به صورت نمایی زیاد است و این تفاوت مهم این مسئله با مسئله دویچ است. برای آنکه تعداد توابع متوازن را بشماریم دقت می کنیم که تعداد ورودی های مختلف این تابع و در نتیجه خروجی های آن برابر است با 2^n . تعداد توابع متوازن برابر است با تعداد طرعی که می توان نیمی از 2^n خروجی را برابر با صفر و نیمی دیگر را برابر با ۱ گرفت و این تعداد برابر است با $\frac{2^n!}{(2^{n-1}!)^2}$ و این عددی است که رابطه اش با n یک رابطه نمایی است. بنابراین تعداد کل توابع ممکن که می توانند ثابت یا متوازن باشند برابر است با $2 + \frac{2^n!}{(2^{n-1}!)^2}$. سوال این است که چند بار می بایست تابع را فراخوانیم تا بتوانیم با اطمینان بگوییم که تابع ثابت است یا متوازن. اگر بخواهیم یک الگوریتم تعینی (یعنی الگوریتمی که با قطعیت پاسخ را مشخص کند) برای این مسئله به کار ببریم باید به ترتیب زیر عمل کنیم. اعداد n بیتی $x := (x_{n-1}x_{n-2} \dots x_1x_0)$ را یک به یک می دهیم (تابع را فرامی خوانیم تا $f(x)$ را محاسبه کند).

هرگاه به ازای دو مقدار مختلف x مقادیر متفاوتی برای $f(x)$ بدست بیاوریم الگوریتم را متوقف کرده و حکم می کنیم که تابع متوازن است. اما اگر چنین نبود الگوریتم را ادامه می دهیم تا نیمی از اعداد بعلاوه یک یعنی $2^{n-1} + 1$ عدد خوانده شوند. اگر برای این تعداد ورودی نیز تابع مقدار یکسان داشته باشد حتماً تابع ثابت است. حال نشان می دهیم که با استفاده از یک الگوریتم کوانتومی می توان تنها با یک بار فراخوانی تابع، متوازن بودن یا ثابت بودن آن را با قطعیت تعیین کرد.

برای این کار از همان شکل ۲ استفاده می کنیم. دقت می کنیم که خروجی بالایی این جعبه سیاه که شامل n تا کیوبیت است در حالت $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle$ است. حال اگر تابع f یک تابع ثابت باشد، آنگاه این حالت برابر است با $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle$ و عمل گیت های هادامارد در سمت راست مدار، روی این حالت حالت $|0 \dots 0\rangle$ را تولید خواهد کرد. اما اگر تابع f یک تابع متوازن باشد، ثابت می

کنیم که عمل گیت های هادامارد، حالتی را تولید خواهد کرد که اندازه گیری اش به هیچ وجه حالت $|000000\rangle$ را تولید نخواهد کرد. برای

اینکه این نکته را بفهمیم باید ثابت کنیم که برای یک تابع متوازن

$$\langle 000000 | H^{\otimes n} | \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} | x \rangle = 0. \quad (11)$$

برای اثبات این رابطه $H^{\otimes n}$ را روی حالت سمت چپ اثر می دهیم. در نتیجه سمت چپ عبارت فوق برابر می شود با

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} = 0. \quad (12)$$

بنابراین ثابت کرده ایم که برای یک تابع ثابت، n کیوبیت اول در حالت $|000000\rangle$ قرار می گیرند و برای یک تابع متوازن این n کیوبیت در هر حالتی غیر از صفر قرار خواهند گرفت. بنابراین با یک بار اندازه گیری این حافظه ی n کیوبیتی می توانیم ثابت بودن یا متوازن بودن یک تابع را تشخیص دهیم.

تمرین: نشان دهید که با استفاده از الگوریتم های تصادفی می توان مسئله دویچ-جوزا را در زمان چندجمله ای حل کرد. برای این کار k را تابع را فراخوانی کنید. اگر در همه موارد مقدار تابع یکسان بود اعلام کنید که تابع ثابت است و در غیر این صورت اعلام کنید که تابع متوازن است. برای آنکه خطای شما در این الگوریتم تصادفی به طور متوسط از ϵ کمتر باشد، مقدار کمینه ی k را پیدا کنید.

تمرین: فرض کنید که بجای کیوبیت ها با کیوتریت ها سرو کار دارید. جعبه سیاه شما نیز تابعی را حساب می کند که سه مقدار ۰، ۱ و ۲ را محاسبه می کند. قول داده شده است که تابع f یا ثابت است یا متوازن. در این جا متوازن بودن به این معناست که تعداد ۰ ها و ۱ ها و ۲ ها در خروجی تابع باهم مساوی است. الگوریتم دویچ و الگوریتم دویچ-جوزا را به این حالت تعمیم دهید.

تمرین: در مسئله قبلی اگر تابع از n کیوتریت به یک کیوتریت تعریف شده باشد، و ما بخواهیم یک الگوریتم تصادفی کلاسیک را به کار ببریم حساب کنید برای اینکه احتمال اشتباه ما به طور متوسط از ϵ کمتر باشد چند بار می بایست تابع را فراخوانی کنیم.

۵ الگوریتم کوانتومی برای مسئله برنشتاین - وزیرانی

این مسئله یک مسئله جعبه سیاه است با این تفاوت که قول داده شده است که تابع به شکل زیر است:

$$f : B_n \longrightarrow B_1, \quad f(\mathbf{x}) := \mathbf{x} \cdot \mathbf{a}, \quad \text{Mod } 2, \quad (13)$$

که در آن

$$a.x = \sum_{k=0}^{n-1} a_k x_k \quad \text{mod } 2. \quad (14)$$

سوال این است که با چند بار فراخواندن تابع می توان مقدار a را تعیین کرد. با الگوریتم های کلاسیک واضح است که می بایست n بار تابع را فراخوانی کرد تا a تعیین شود. نشان می دهیم که با یک الگوریتم کوانتومی می توان تنها با یک بار فراخوانی مقدار a را مشخص کرد. برای این الگوریتم باز هم از همان مدار شکل ۲ استفاده می کنیم. می دانیم که در این مدار حالت زیر تولید می شود.

$$|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{a.x} |x\rangle. \quad (15)$$

حال اثر عملگرهای هادامارد این حالت را تبدیل به حالت زیر می کند:

$$|\phi'\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{x.(a+y)} |x\rangle. \quad (16)$$

اما می دانیم که

$$\sum_{x=0}^{2^n-1} (-1)^{(a+y).x} = 2^n \delta_{a,y}. \quad (17)$$

بنابراین خواهیم داشت $|\phi'\rangle = |a\rangle$ ، یعنی در حقیقت مقدار a تنها با یک بار اثر کردن تابع در n کیوبیت بالا نوشته خواهد شد.

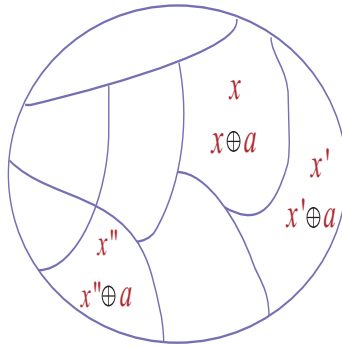
تمرین: فرض کنید که بجای کیوبیت ها با کیوتریت ها سرو کار دارید. جعبه سیاه شما نیز تابعی را حساب می کند که سه مقدار ۰، ۱ و ۲ را محاسبه می کند. قول داده شده است که تابع f به این صورت است که

$$f(\mathbf{x}) := \mathbf{x} \cdot \mathbf{a}, \quad \text{Mod } 3. \quad (18)$$

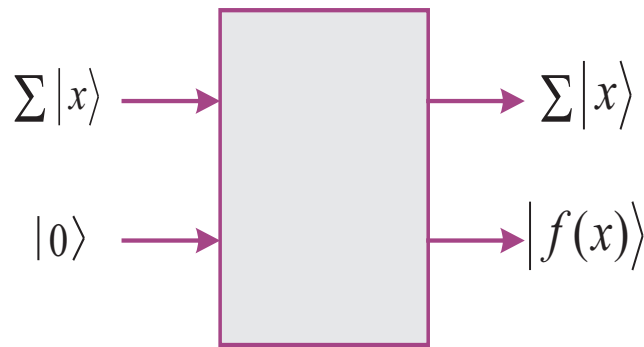
الگوریتم برنشتاین - وزیرانی را طوری تعمیم دهید که این مسئله را با یک بار فراخوانی تابع حل کند.

۶ الگوریتم کوانتومی برای مسئله سیمون

حال به مسئله سیمون *Simon* می پردازیم که با مسئله های قبلی تفاوت زیادی دارد. در حقیقت تعمیم این مسئله به الگوریتم شر *Shor* منجر می شود که در درس های آینده به آن خواهیم پرداخت.



شکل ۲: مسئله سیمون



شکل ۳: ورودی و خروجی در مسئله سیمون. حالت خروجی درهم تنیده است.

تابع $f : B_n \rightarrow B_m$ را در نظر بگیرید. در این مسئله شرطی که تابع در آن صدق می کند این است که تمامی مجموعه B_n که شامل 2^n عضو است به 2^{n-1} زیر مجموعه دو عضوی تقسیم شده است. هر زیر مجموعه دارای دو عضو x و $x \oplus a$ است. دقت کنید که a و x هر دو اعداد n بیتی هستند و عمل $\oplus$ بیت به بیت انجام می شود. تابع f روی هر زیر مجموعه دو عضوی یک مقدار دارد، یعنی $f(x) = f(x \oplus a)$ و مقدار تابع روی هیچ دو زیر مجموعه ای باهم برابر نیست. می خواهیم با فراخوانی تابع مقدار a را پیدا کنیم. حال سوال این است که با چند بار خواندن تابع می توانیم مقدار a را تعیین کنیم. برای این مسئله می بایست از یک مدار کوانتومی متفاوت استفاده کنیم. این مدار در شکل ۶ نشان داده شده است.

حالت ورودی در این الگوریتم عبارت است از:

$$|\Psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, 0\rangle \quad (19)$$

حالت خروجی برابر است با حالت زیر که یک حالت درهم تنیده بین حافظه های بالا و پایین در مدار شکل ۶ است.

$$|\Psi_r\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, f(x)\rangle \quad (20)$$

مرحله بعدی آن است که حافظه دوم یعنی آن قسمتی که $f(x)$ را در بردارد اندازه گیری کنیم. فرض کنید که مقدار $f(x)$ بدست آید. در این صورت حافظه اول به حالت زیر تقلیل پیدا می کند:

$$|\Phi_1\rangle = \frac{1}{\sqrt{2}}(|x.\rangle + |x. \oplus a\rangle). \quad (21)$$

اگر این حافظه را اندازه گیری کنیم مقدار a مشخص نمی شود زیرا با احتمال مساوی یکی از مقادیر x یا $x \oplus a$ پدیدار خواهند شد که از هیچ کدام نمی توان مقدار a را بدست آورد.

در مرحله بعدی روی این حالت گیت های هادامارد $H^{\otimes n}$ را اثر می دهیم. حالتی که بدست می آید برابر است با:

$$|\Phi_r\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} [(-1)^{x \cdot a}] |y\rangle. \quad (22)$$

با کمی محاسبه خواننده می تواند نشان دهد که این حالت برابر است با:

$$|\Phi_r\rangle = \frac{1}{\sqrt{2^{n-1}}} \sum_{a \cdot y=0} [(-1)^{x \cdot y}] |y\rangle. \quad (23)$$

این امر به این معناست که هرگاه ما حالت $|\Phi_r\rangle$ را اندازه بگیریم، عددی مثل y بدست می آید که در رابطه $a \cdot y = 0$ صدق می کند. با حدود n بار اندازه گیری می توانیم تعداد n معادله مستقل برای a پیدا کنیم که حل کردن آنها عدد a را بدست می دهد.

باید دقت کنیم که هیچ کدام از الگوریتم های معرفی شده در این درس کلاس حل پذیری مسئله را تغییر نداده اند. یعنی نتوانسته اند مسئله ای را از کلاس نمایی به کلاس چند جمله ای تبدیل کنند. اما این الگوریتم ها مقدمه ای خواهند بود برای الگوریتم بسیار مهم شُر^۷ که در درس های آینده با آن آشنا خواهیم شد. این الگوریتم تنها الگوریتم کوانتومی شناخته شده است که می تواند کلاس حل پذیری یک مسئله یعنی مسئله تجزیه اعداد را تغییر دهد.

تمرین: فرض کنید که با کیوتریت ها سرو کار دارید و زیر مجموعه های دوعضوی در مسئله سیمون سه عضوی هستند به این معنا که هر زیر مجموعه دارای سه عضو x ، $x \oplus a$ و $x \oplus a \oplus a$ است. نشان دهید که بازهم می توانید مسئله سیمون را با یک الگوریتم کوانتومی حل کنید.

