Computação Quântica

`http://www.if.ufrgs.br/~jgallas`

Conteúdo

Bibliografia

Fontes básicas, de onde o material destas notas foi extraído primordialmente :

1. *The Physics of Quantum Information*, D. Bouwmeester, A. Ekert and A. Zeilinger, editors, Springer, Heidelberg, 2000.
2. Material contido em, por exemplo, `www.qubit.org` e `www.quantum.at`
3. *Feynman Lectures on Computation*, R.P. Feynman, Penguin Books, London, 1996.
4. *Explorations in Quantum Computing*, C.P. Williams and S.H. Clearwater, Springer-TELOS, NY, 1998.

2 Tópicos de Mecânica Quântica e Computadores

2.1 Estados de um Registro Quântico de Memória

Já vimos como descrever um sistema contendo apenas dois estados (qubits): usando-se os estados

$$|0\rangle \equiv \begin{pmatrix}1\\0\end{pmatrix}, \qquad |1\rangle \equiv \begin{pmatrix}0\\1\end{pmatrix}, \tag{1}$$

que formam uma *base* do sistema.

Entretanto, um registro de memória consiste de *vários* sistemas com dois estados. Como se pode descrever os estados de um tal registro? Através de um *produto direto*.

Por exemplo, para *dois* sistemas, cada um com *dois* estados, escrevemos

$$|0\rangle_1\,, \quad |1\rangle_1\,, \quad |0\rangle_2\,, \quad |1\rangle_2\,, \tag{2}$$

onde $|0\rangle_j$ e $|1\rangle_j$, representam os estados individuais (iniciais) de cada um dos dois sistemas $j = 1, 2$.

Usando-se tal base, os estados arbitrários genéricos de cada um dos dois sistemas são representados do seguinte modo:

$$|\Psi\rangle_1 = a\,|0\rangle_1 + b\,|1\rangle_1 \equiv \begin{pmatrix}a\\b\end{pmatrix}, \tag{3}$$

$$|\Psi\rangle_2 = A\,|0\rangle_2 + B\,|1\rangle_2 \equiv \begin{pmatrix}A\\B\end{pmatrix}. \tag{4}$$

Após estas preliminares estamos preparados para construir a representação (base) matemática que será utilizada para representar os estados de um regitro de memória formado com um número arbitrário de sistemas com dois qubits. Eis aqui o caso mais simples.

O estado arbitrário de um registro de memória formado por dois sistemas contendo dois qubits cada é definido pelo chamado *produto direto*

$$|\Psi\rangle_1 \otimes |\Psi\rangle_2 = \begin{pmatrix}a\\b\end{pmatrix} \otimes \begin{pmatrix}A\\B\end{pmatrix} \equiv \begin{pmatrix}aA\\aB\\bA\\bB\end{pmatrix}. \tag{5}$$

Com esta definição podemos encontrar facilmente os quatro estados que formam a *base* de uma memória construida utilizando-se dois sistemas de dois qubits:

$$|00\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \equiv \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \qquad |01\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} \equiv \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, \tag{6a}$$

$$|10\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \equiv \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, \qquad |11\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} \equiv \begin{pmatrix}0\\0\\0\\1\end{pmatrix}. \tag{6b}$$

O estado mais geral de um registro de memória formado por dois sistemas de dois qubits é, portanto,

$$|\psi\rangle = \Psi_{00}\,|00\rangle + \Psi_{01}\,|01\rangle + \Psi_{10}\,|10\rangle + \Psi_{11}\,|11\rangle = \begin{pmatrix}\Psi_{00}\\\Psi_{01}\\\Psi_{10}\\\Psi_{11}\end{pmatrix}. \tag{7}$$

A generalização para registros de memória contendo *vários* sistemas de dois qubits é simples.

Uma das propriedades mais importante dos registros quânticos de memória é a sua capacidade de armazenar *quantidades exponenciais* de informação clássica utilizando apenas uma *quantidade polinomial* de qubits, explorando-se o princípio de superposição.

Em particular, é possível fazer-se a evolução do conjunto de qubits de um registro quântico de memória para um estado que *não* pode ser expressado matematicamente através de um produto direto de estados dos qubits individuais que compõem o registro.

Num tal estado entrelaçado o registro de memória está, como um todo, num estado quântico bem definido mesmo embora os qubits que o compõem não o estejam!

Se tentassemos ler um registro quântico de memória que estivesse num tal estado entrelaçado, através de medidas feitas sobre um qubit de cada vez, veríamos então que os valores que obteríamos para os últimos qubits da seqüência seriam determinados pelos resultados das medidas feitas nos primeiros qubits da seqüência.

Portanto, se um registro quântico de memória estiver num estado entrelaçado, poderemos alterar o estado em uma parte do registro simplesmente fazendo medições noutra parte do registro.

Este fenômeno é importante em muitos algorítmos quânticos e não tem paralelo em física clássica.

Mais adiante mostraremos como os estados entrelaçados podem ser explorados na teleportação, na transmissão de informação e códigos error-correcting.

2.2 Operadores Unitários

A descrição matemática de como 'algo' muda no mundo quântico é dado pela aplicação de um *operador*.

Um operador é representado por uma matriz quadrada.

Se o sistema quântico consiste de n sistemas de dois estados quânticos (como é o caso do registro de memória de um computador quântico), então qualquer operador que atua no registro de memória seria representado por uma matriz de dimensão $2^n \times 2^n$.

Assim sendo, a forma mais geral para um operador que atue num *único* qubit é uma matriz 2×2:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \omega_0 \\ \omega_1 \end{pmatrix} = \begin{pmatrix} a\omega_0 + b\omega_1 \\ c\omega_0 + d\omega_1 \end{pmatrix}. \tag{8}$$

Eis aqui outro operador, um que transforma um autoestado em uma superposição (verifique isto!):

$$U(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{9}$$

Para obter um superposição com pesos iguais, giramos um autovetor de um ângulo de $\pi/4$, por exemplo,

$$U(\frac{\pi}{4})\ |0\rangle = U(\frac{\pi}{4})\ \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \frac{1}{\sqrt{2}}\Big[\,|0\rangle + |1\rangle\,\Big]. \tag{10}$$

Analogamente, pode-se pensar nas portas lógicas como sendo operadores. O operador de negação, NOT, definido como

$$\text{NOT} \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{11}$$

tem o efeito de *negar* ("flipar") o estado da sua entrada, do seguinte modo;

$$\text{NOT}\ |0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle\,, \tag{12}$$

$$\text{NOT}\ |1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle\,. \tag{13}$$

Note que os operadores $U(\theta)$ e NOT são invertíveis, no sentido que se pode inferir a entrada partindo-se da saida, e vice-versa.

De fato, conforme veremos a seguir, a evolução de qualquer sistema quântico *não-observado* é sempre descrita por operadores invertíveis.

Em termos matemáticos, este operadores são descritos por matrizes unitárias. A propriedade que define uma matriz unitária é que sua transposta conjugada é igual a sua inversa.

Voltemos à computação.

Um conjunto de portas lógicas é dito ser "universal" se qualquer computação factível pode ser efetuada num circuito contendo apenas portas do tipo encontradas no conjunto.

Em computação clássica é bem sabido que existem muitos conjuntos universais de portas.

Por exemplo, qualquer computação clássica pode ser feita usando um circuito construido com portas NOT e AND.

Um resultado semelhante é válido para computação quântica.

Adriano Barenco e seus colaboradores[1] mostraram que uma porta de dois qubits que tenha a forma

$$A(\phi,\alpha,\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\alpha}\cos\,\theta & -ie^{i(\alpha-\phi)}\sin\,\theta \\ 0 & 0 & -ie^{i(\alpha+\phi)}\sin\,\theta & e^{i\alpha}\cos\,\theta \end{pmatrix} \tag{14}$$

onde ϕ, α a θ são constantes irracionais múltiplas de π e uma outra, é universal para computação quântica. A mesmo conclusão foi alcançada por David DiVincenzo[2], independentemente.

Portanto, qualquer computação quântica factível pode ser efetuada conectando-se apenas portas da forma $A(\phi,\alpha,\theta)$ num circuito quântico.

2.3 O Computador Quântico de Benioff

Benioff[3] introduziu a idéia que a "fita" da máquina de Turing poderia ser substituida por uma seqüência de sistemas quânticos simples com dois estados. Tal idéia forneceu um meio primitivo de codificar uma seqüência de dígitos binários. Analogamente, a "cabeça" da máquina de Turing foi substituida por uma interação quântica que podia ler e/ou resetar o valor do estado do spin. As "regras" da máquina de Turing foram substituidas por uma equação de Schrödinger cuidadosamente desenhada de modo que uma configuração inicial de spins evoluisse para um estado final que, quando codificado como bits, seria interpretável como tendo executado algum cálculo útil. Portanto, o "programa" que o computador estava executando estava contido implicitamente nos detalhes da equação de Schrödinger. A máquina evoluia em passos de duração fixa tais que, ao final de cada passo, a fita estava novamente em um de seus estados fundamentais no qual cada spin estava ou totalmente "up" (representando um 1) ou totalmente "down", representando um 0. Entretanto, *durante* um passo a máquina poderia ir temporariamente para uma superposição de estados.

Assim, o computador de Benioff deixou de fazer uso completo do potencial das computações superpostas porque ao concluir-se cada passo computacional a cabeça media o estado da fita que, deste modo, colapsava quaisquer superposições que porventura existissem na fita. Consequentemente, coerência quântica, e portanto interferência entre computações superpostas, a marca registrada das computações quânticas verdadeiras, era destruida ao final de cada passo. Apesar disto, foi a primeira vez que alguém havia demonstrado que interações de natureza quântica poderiam ser utilizadas para fazer uma mímica das operações duma máquina de Turing.

O computador de Benioff não é prático como projeto de um computador real pois depende de interações entre a "cabeça" e os spins da "fita" que poderiam estar muito separados. Em computadores reais seria melhor ter-se que lidar apenas com spins próximos. Pior ainda, para efetivamente estabelecer o Hamiltoniano necessário para fazer-se uma mímica de uma máquina de Turing particular, teria-se que saber, não apenas o programa mas também o conjunto completo de órbitas computacionais. Em outras palavras, teria-se que saber a resposta ao problema que estivessemos tentando computar de modo a poder desenhar o computador quântico para poder resolve-lo! Afortunadamente, através de refinamentos do seu modelo original, Benioff mostrou que tal situação pode ser remediada usando-se um Hamiltoniano dependente do tempo, embora a questão relativa aos spins remotos não o possa[4, 5, 6, 7].

Em 1985, Feynman[8] descobriu um modo geral de transformar o modelo de Benioff num que apenas necessitava spins locais.

2.4 O Computador Quântico de Feynman

O modelo da máquina de Turing é um modo de descrever um computador abstrato. Um outro modo, é com um circuito construido a partir de um conjunto de portas lógicas primitivas. Os dois modos são equivalentes: qualquer computação que pode ser feita eficientemente com uma máquina de Turing pode também ser feita por um circuito quântico, e vice-versa[9].

O modelo de Feynman de um computador quântico é mais como uma versão quântica de um circuito de lógica combinatorial. Começa-se com uma descrição a nível de circuito da computação a ser efetuada. O circuito é construido com portas quânticas reversíveis, tais como as portas de 2-qubits de Barenco, especificadas anteriormente[1]. Em geral, podemos considerar o circuito como consistindo de k portas lógicas atuando em m qubits. A transformação unitária global realizada pelo circuito pode portanto ser escrita como $A_k \cdot A_{k-1} \cdot \cdots \cdot A_1$ onde A_i é o operador que descreve a ação da i-ésima porta.

Portanto, a questão passa a ser qual o Hamiltoniano $\hat{H}$ na expressão

$$\hat{\mathcal{U}}(t) = \exp\Big(-\frac{i}{\hbar}\hat{H}t\Big) = \Big(\exp(-\frac{i}{\hbar}\hat{H})\Big)^t \tag{15}$$

irá dar origem à evolução dinâmica que implementa $A_k \cdot A_{k-1} \cdot \cdots \cdot A_1$.

Esta é a questão para a qual Feynman encontrou uma resposta geral.

Ele o fez através de um aumento do conjunto dos m qubits (usado para representar as entradas e saidas do circuito) com um conjunto de $k+1$ qubits extra.

Estes qubits foram usados meramente para registrar-se o progresso da computação; ou seja, quantas portas haviam sido utilizado até então.

Os qubits extras constituiam, portanto, uma espécie de contador de passos do programa. Matematicamente, isto pode ser feito escrevendo-se o Hamiltoniano como

$$\hat{H} = \sum_{i=1}^{k} c_{i+1} \cdot a_i \cdot A_{i+1} + \Big(c_{i+1} \cdot a_i \cdot A_{i+1}\Big)^\dagger, \tag{16}$$

onde c e a são operadores de criação e aniquilação, respectivamente (veja o capítulo 3).

Apenas um dos sítios contadores do programa, chamado de "cursor" irá estar ocupado. Portanto, medindo-se periodicamente o cursor podemos determinar quando todas as k operações das portas foram aplicadas. Tão logo encontremos o cursor no k-ésimo sítio, sabemos que o circuito inteiro foi aplicado à entrada. Neste instante o estado dos m qubits contém garantidamente uma resposta válida à computação representada pelo circuito.

Desta forma o Hamiltoniano irá fazer o computador evoluir para uma superposição de estados representando números diferentes de aplicações de portas. Portanto, o cursor também ocupa um estado superposto e o tempo para completar a tarefa fica indefinido.

Existe entretanto uma conexão importante entre a independência temporal do Hamiltoniano e a localidade do operador unitário de evolução. Como $\hat{H}$ e $\hat{\mathcal{U}}(t)$ estão relacionados por uma exponencial, podemos supor expandir-se a exponencial numa série de potências em $\hat{H}$. Assim, na verdade $\hat{\mathcal{U}}(t)$ envolve uma soma de matrizes, potências de $\hat{H}$. Portanto se $\hat{H}$ envolve interações apenas entre qubits adjacentes no registro de memória, então $\hat{\mathcal{U}}(t)$ irá envolver interações entre qubits que estão arbitrariamente longe uma vez que o efeito das potências da matriz é aumentar acoplamentos a qubits mais remotos. Isto é considerado indesejável pois pode ser difícil de se conseguir fisicamente.

2.5 O Computador Quântico de Deutsch

A primeira máquina quântica de Turing verdadeira foi idealizada em 1985 por David Deutsch, da Universidade de Oxford[10].

O modelo de Deutsch diferia do modelo de Benioff pelo fato de manter o registro quântico de memória (a "fita") numa superposição de estados durante toda a operação do computador.

No computador quântico de Deutsch deseja-se que o operador $\hat{U}$, de evolução temporal, seja local (i.e. apenas envolva interações entre qubits adjacentes) e seja independente do tempo.

Isto assegura que, a cada passo computacional, apenas qubits adjacentes interajam e que a computação tenha um tempo de duração bem definido.

Porém, uma evolução unitária fixa e local não pode ser conseguida com um Hamiltoniano fixo local.

É necessário ter-se um Hamiltoniano dependente do tempo para atingir tal objetivo.

A solução da equação de Schrödinger para o caso de um Hamiltoniano dependente do tempo pode ser escrita assim:

$$|\Psi(t)\rangle = T\left\{\exp\Big(-\frac{i}{\hbar}\int_0^t \hat{H}(\tau)\,d\tau\Big)\right\}|\Psi(0)\rangle \tag{17}$$

onde T é chamada "integral ordenada no tempo", definida como[11]

$$T\left\{\exp\Big(-\frac{i}{\hbar}\int_0^t \hat{H}(\tau)\,d\tau\Big)\right\} = \lim_{N\to\infty}\prod_{n=0}^{N-1}\exp\Big(-\frac{i}{\hbar}\hat{H}(n\Delta)\,\Delta\Big), \tag{18}$$

onde $\Delta = t/N$.

O operador unitário de evolução $\hat{U}$ pode ser tornado local e independente do tempo através da escolha de um Hamiltoniano com a forma

$$\hat{H}(t) = \frac{-\pi\, f(t)}{2T}\begin{pmatrix} 0 & & & & & & & \\ & 0 & & & & & & \\ & & 0 & & & & & \\ & & & 0 & & & & \\ & & & & 0 & & & \\ & & & & & 0 & & \\ & & & & & & 1 & \alpha \\ & & & & & & -\alpha & 1 \end{pmatrix} \tag{19}$$

onde $f(t)$ é uma função qualquer tal que:

$$\frac{1}{T}\int_0^T f(t)\,dt = 1, \tag{20a}$$

$$\lim_{t\to 0} f(t) = 0, \tag{20b}$$

$$\lim_{t\to T} f(t) = 0, \tag{20c}$$

onde os limites devem ser atingidos de modo suave.

Neste curso faremos apenas simulação do computador mais simples, o de Feynman, a partir do Capítulo seguinte.

Referências

[1] A. Barenco, *A universal two-bit gate for quantum computation*, Proc. R. Soc. London, **449A**, 679-683(1995).

[2] D. DiVincenzo, *Two-bit gates are universal for quantum computation*, Phys. Rev. A, **51**, 1015-1022(1995).

[3] P. Benioff, *The computer as a physical system: a microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines*, J. Stat. Phys., **22**, 563-591(1980).

[4] P. Benioff, *Quantum mechanical Hamiltonian models of discrete processes*, J. Math. Phys., **22**, 495-507(1981).

[5] P. Benioff, *Quantum mechanical models of Turing machines that dissipate energy*, Phys. Rev. Lett., **48**, 1581-1585(1982).

[6] P. Benioff, *Quantum mechanical Hamiltonian models of discrete processes that erase their own histories: application to Turing machines*, Intern. J. Theor. Phys., **21**, 177-202(1982).

[7] P. Benioff, *Quantum mechanical Hamiltonian models of computers*, Ann. N.Y. Acad. Science, **480**, 475-486(1986).

[8] R.P. Feynman, *Quantum mechanical computers*, Optics News, **11**, 11-20(1985). Este artigo aparece como "Capítulo 6" no livro do Feynman, *Lectures on Computation*, mencionado no início.

[9] A. Yao, *Quantum circuit complexity*, Proceedings of the 34th IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, págs. 352-360, 1993.

[10] D. Deutsch, *Quantum theory, the Church-Turing principle, and the Universal Quantum Computer*, Proc. Royal Soc. London, **400A**, 97-117(1985).

[11] R. Shankar, *Principles of Quantum Mechanics*, second edition, Plenun Press, New York, 1994, página 148.