

Computação Quântica

Arthur E., Nascimento, Bruno S. Martin, Robinson M. Nakamura

Instituto de Computação – Universidade Estadual de Campinas, Brazil

[arthuremnascimento, robmnk, brunosmartin]@gmail.com

***Abstract.** This work aims to make a brief introduction to the quantum computing. We will introduce some basic concepts of quantum mechanics as superposition, the entanglement and the principle of uncertainty. We'll tell about general functioning of quantum computers using the qubits. Discussing a bit of quantum complexity, quantum algorithms and what the advantages of this new paradigm of computing. We must remember that we won't get deep in any of the topics to keep the brevity of the discussion.*

***Resumo.** Este trabalho tem como objetivo fazer uma breve introdução sobre a computação quântica. Iremos introduzir alguns conceitos básicos sobre mecânica quântica como a superposição, o entrelaçamento e o princípio da incerteza. Falaremos sobre funcionamento geral dos computadores quânticos utilizando os qubits. Discutiremos um pouco de complexidade quântica, algoritmos quânticos e quais as vantagens deste novo paradigma de computação. Devemos lembrar que não nos aprofundaremos muito em nenhum dos tópicos para manter a brevidade da discussão.*

1. Introdução

O matemático Inglês Alan Turing na década de 30 criou um modelo computacional abstrato que se tornou um paradigma de computação conhecido como Máquina de Turing. Simplificadamente, uma máquina de Turing é uma idealização que opera com seqüências lógicas de unidade de informação que conhecemos como bits (do inglês binary digit). Os microcomputadores que temos hoje em nossas casas é uma idealização física da máquina de Turing.

No ano de 1970, um dos fundadores da empresa fabricante de microprocessadores Intel, Gordon Moore, notou um aumento vertiginoso no número de componentes por unidade de volume no chips ao longo dos anos e, como não podia deixar de ser, uma redução no tamanho físico dos bits. Traduzindo em números de átomos necessários para representar um bit de informação, podemos ter uma idéia dessa redução: em 1950 eram necessários cerca de 10^{19} (10 elevado a 19) átomos para representar um bit. Atualmente são "apenas" cerca de 10^9 (10 elevado a 9), uma redução de 10 ordens de magnitude! Se aplicarmos a Lei de Moore e fizermos uma projeção sobre os próximos vinte anos, o resultado é algo espantoso: em 2020, um bit de informação será representado por apenas 1 único átomo! [Oliveira, 2002].

Um átomo representando 1 bit poderia significar o limite físico dos computadores, pois seria a densidade de bits máxima por chip. No entanto, há algo mais do que uma simples limitação física de memória para 2020, de acordo com a Lei de Moore, e a razão é direta e simples: na escala atômica, as idéias clássica não são válidas, mas sim a Mecânica Quântica. Portanto os processos computacionais devem obedecer às leis dessa teoria física.

No mundo dos átomos e moléculas as leis de Newton não funcionam. Para descrever corretamente o comportamento dos objetos moleculares, é preciso utilizar as leis da Mecânica Quântica. As diferenças entre estas leis tem conseqüências dramáticas para a computação, e a razão é a seguinte: os circuitos eletrônicos que representam os bits de informação nos computadores atuais são objetos clássicos, e portanto seguem as leis da física clássica. Como conseqüência, cada bit em um computador clássico só pode adquirir um dos valores, "0" ou "1", que são, por sua vez, mutuamente excludentes. Acontece que no mundo dos átomos, a Mecânica Quântica nos ensina que os bits (que são chamados de *quantum-bits*, ou *qubits* no mundo quântico) podem simultaneamente adquirir os valores "0" e "1"! Esta surpreendente propriedade é chamada de *superposição de estados quânticos*.

A propriedade da superposição já foi demonstrada muitas vezes em laboratórios de física em todas as partes do mundo, e é uma verdade incontestável. Pensando em computação, ela representa um ganho inimaginável de velocidade de processamento, pois todas as seqüências de bits possíveis em um computador poderiam ser manipuladas simultaneamente. Em 1993 um cientista chamado Peter Shor executou uma das mais espetaculares demonstrações de ganho de velocidade ao inventar um algoritmo quântico para fatorar números grandes, que é um problema difícil para a computação clássica. Na tabela abaixo há uma comparação, em função do número a ser fatorado, entre os tempos necessários de fatoração entre algoritmos clássicos e o algoritmo de Shor. [Oliveira, 2002].

Tabela 1. Comparativo entre o tempo de execução da fatoração de inteiros grandes utilizando abordagem clássica e o algoritmo de Shor.

Comprimento do número a ser fatorado (em bits)	Tempo de fatoração por algoritmo clássico	Tempo de fatoração com o algoritmo de Shor
512	4 dias	34 segundos
1024	100 mil anos	4,5 minutos
2048	100 mil bilhões de anos	36 minutos
4096	100 bilhões de quatrilhões de anos	4,8 horas

A segurança de mensagens criptografadas popularmente utilizada na Internet, como números de cartões de crédito, é baseada na fatoração de números grandes. Fica portanto demonstrado que no dia que um computador quântico estiver em funcionamento as mensagens classicamente criptografadas, nunca mais serão secretas, de acordo com o algoritmo de Shor.

2. Mecânica Quântica

A Teoria Quântica foi o maior avanço da física no século XX, provocando uma revolução científica. A mecânica quântica possui características probabilísticas, ao contrário da mecânica clássica. Coloquialmente, costuma-se descrever a mecânica quântica como uma teoria na qual nada é o que parece. De acordo com o principio da incerteza, que advem da teoria quântica, nada pode ser medido tão precisamente quanto se deseja, pois a medição sempre afeta o estado do que foi medido. Portanto esta teoria é formulada precisamente com base em uma estrutura matemática coerente.

Da teoria quântica ainda surge o princípio da dualidade partícula-onda. Segundo este princípio, um elétron, por exemplo, pode comportar-se como partícula e as vezes como onda.

Por outro lado, toda onda possui uma partícula associada. O físico Richard Feynman usava um bom exemplo para explicar esta questão. Imagine luz sendo refletida por um espelho. Nenhum espelho é perfeito, deste modo apenas 95 % desta luz é refletida pelo espelho e os outros 5% o atravessam, ou é absorvido ou perdido.

Classicamente esta era uma situação completamente aceitável. Porém, sabe-se, da descoberta de Planck, que a luz é dividida em pacotes, ou quanta, chamados fótons. Estes fótons são indivisíveis. Desta forma, um fóton deve ser completamente absorvido ou refletido. Não é possível que um fóton seja parcialmente refletido e parcialmente absorvido. Então, conclui-se que 19 fótons de 20 são refletidos pelo espelho e o outro é absorvido. Mas como saber qual é absorvido e quais são refletidos?

Não é possível saber. Um fóton tem 95% de chance de ser refletido e 5% de chance de ser absorvido. Não há nenhuma regra ou propriedade secreta do fóton que possa prever seu comportamento. A imprevisibilidade é inata.

3. Qubits e Registradores Quânticos

Para se ter uma definição rigorosa de *qubit* são necessários conceitos matemáticos avançados que vão além dos objetivos deste artigo, de forma que serão ilustradas algumas de suas propriedades para compreender o funcionamento do Computador Quântico. O *qubit* pode assumir, além dos estados 0 e 1, uma superposição dos dois estados em um dado instante. Seu estado pode ser representado por $\alpha|0\rangle + \beta|1\rangle$ para qualquer α e β tal que $|\alpha|^2 + |\beta|^2 = 1$, onde $|\alpha|^2$ é a probabilidade do *qubit* estar no estado $|0\rangle$ e $|\beta|^2$ é a probabilidade de estar no estado $|1\rangle$. O *qubit* pode ser fisicamente implementado de diversas formas, utilizando-se um átomo, um *spin* nuclear ou de elétron, um fóton polarizado e até mesmo utilizando supercondutores.

Podemos agrupar os *qubit* para formar os registradores quânticos, que também podem assumir estados superpostos. Enquanto um registrador clássico de n bits só pode assumir um estado em determinado momento, o registrador quântico pode assumir uma superposição do 2^n estados clássicos possíveis.

Um registrador quântico de 3 *qbit*, por exemplo, pode ser descrito pela equação

$$|\psi\rangle = a|000\rangle + b|001\rangle + c|010\rangle + d|011\rangle + e|100\rangle + f|101\rangle + g|110\rangle + h|111\rangle \quad (1)$$

onde $a, b, c \dots h$ são número complexos, cujo quadrado da amplitude representa a probabilidade de o registrador estar naquele estado. Por exemplo, a probabilidade de o registrador quântico estar no estado 100 é $|e|^2$. Assim, para armazenar o estado de um registrador quântico é necessário um número exponencial de número complexos. Na representação clássica, o número de bits cresce exponencialmente quando se aumentam o numero de *qubit*. Assim, para representar um registrador quântico de tamanho 300, são necessários 10^{90} bits clássico, quantidade maior que a de átomos no universo.

4. Projeto de Computadores Quânticos

Analogamente ao circuitos clássico, os quânticos também são projetados utilizando portas lógicas quântica, que realizam operações elementares com os *qubit*, formando uma rede que implemente a função desejada. Estas portas lógicas devem ser

reversíveis, (i.e. a partir da saída é possível saber a entrada) para garantir as propriedades matemáticas de operação dos *qubits*. As portas lógicas clássicas (como AND e OR) não são reversíveis.

O grande ganho de desempenho do computador quântico está no fato de processar simultaneamente todos os estados possíveis do registrador quântico.

Usando nosso exemplo do tópico anterior de um registrador quântico de 3 *qubits*, a execução de um algoritmo que implemente uma função f processará todos os 8 estados possíveis simultaneamente. Porém não é possível medir estes 8 estados pois, pelas leis da mecânica quântica, isso influenciaria o estado do registrador. O que se tem é uma única leitura ao final da execução do algoritmo. A saída será o estado que estiver associado a maior coeficiente na equação (1) após as transformações realizadas pelo algoritmo, ou seja, o estado que estiver associado a maior probabilidade. Dessa forma, a saída será uma *string* de 3 bits, que pode ser entendida pelo computadores clássico.

Assim, a mudança de paradigma é que o computador quântico é probabilístico, enquanto o computador clássico é determinístico.

5. Complexidade e Algoritmos

5.1. Complexidade Clássica

Sabemos hoje que existem problemas computacionais que são resolvidos de forma eficiente, no pior caso em tempo polinomial no tamanho da entrada como, por exemplo, ordenação e testes de primalidade. Sabemos também que existem problemas computacionais que não conseguem ser resolvidos eficientemente e podem ter complexidade até exponencial e fatorial no tamanho da entrada como, por exemplo, descobrir o caminho hamiltoniano de menor custo em um grafo (caixeiro viajante). E por fim sabemos que existem problemas que não podem ser resolvidos com qualquer complexidade, por pior que seja, são eles os problemas indecidíveis como o problema da parada que diz se um programa para ou executa infinitamente. Os problemas eficientes estão na classe P . Os problemas não eficientes estão na classe NP na qual os problemas mais difíceis estão na subclasse NP -Completo. E os problemas indecidíveis estão fora da classe NP pois nem os algoritmos não-determinísticos conseguem resolvê-los em tempo polinomial no tamanho da entrada.

Outra classe de problemas que devemos citar é a BPP (Bounded-error probabilistic polynomial time) que engloba os problemas que podem ser resolvidos em tempo polinomial com a ajuda de um gerador de números aleatórios (mais formalmente em uma Máquina de Turing probabilística, de acordo com o teorema de Church-Turing moderna) com uma pequena probabilidade de erro. Ou seja, em qualquer execução do algoritmo aleatório tem-se uma pequena probabilidade da resposta estar errada. BPP contém P .

Depois desta breve introdução, nos resta perguntar, o que a computação quântica traz de novidade para a teoria de complexidade? Um algoritmo quântico será capaz de resolver em tempo polinomial problemas que a computação clássica não consegue? E quanto aos problemas indecidíveis, haverá alguma esperança?

5.2. Complexidade Quântica

Segundo o teorema de Church-Turing moderno, qualquer implementação física de modelos de computação pode ser simulado por uma Máquina de Turing probabilística em tempo polinomial no tempo de execução. Há indícios – dentre eles o algoritmo de

Shor que será discutido logo mais— de que computação quântica viole o teorema de Church-Turing moderno no sentido de que seja bem provável que uma Máquina de Turing probabilística não consiga simular uma máquina quântica. Desta forma sente-se a necessidade de ampliar a teoria da complexidade computacional explorando a complexidade de modelos computacionais baseados na mecânica quântica.

Dentre as novas classes de problemas criadas para classificar os problemas quânticos iremos nos concentrar na classe BQP para conseguir fazer um paralelo com a complexidade tradicional. A classe BQP (Bounded-error quantum polynomial time) é a classe de problemas que podem ser resolvidos de forma eficiente por máquinas quânticas com uma pequena probabilidade de erro. É análogo ao BPP.

Acredita-se que $P \subseteq BPP \subseteq BQP$ e esta afirmação é provada mostrando que uma Máquina de Turing probabilística pode ser simulada por uma Máquina de Turing quântica [Vazirani 2002].

Fica mais fácil visualizar a relação entre os conjuntos olhando para a figura 1 [Kaye 2007]:

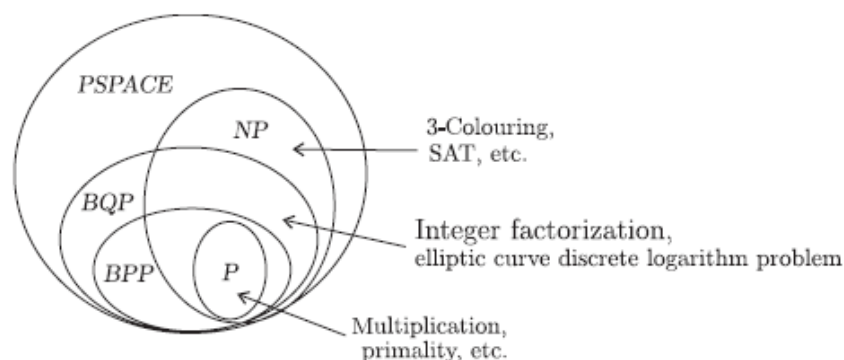


Figura 1. Diagrama de classes.

Vamos analisar agora o que significa BQP conter BPP e conter problemas fora de BPP que estão em NP. Para este fim, vamos dar um exemplo de um problema que esta em NP mas não esta em BPP. É o problema de fatorar inteiros. A dificuldade de se fatorar um número inteiro fica clara quando falamos de números grandes com mais de 100 dígitos. O melhor algoritmo que fatora um inteiro grande é o General Number Field Sieve (GNFS) que tem complexidade $O(e^{n^{1/3} * \log n^{2/3}})$ [Kaye 2007] onde n é o número de bits. Como o algoritmo é exponencial no tamanho da entrada ele esta fora de P. Será que a fatoração de inteiros está em BQP?

5.3.O Algoritmo de Shor

Shor propôs um algoritmo que pode ser considerado um dos maiores avanços na teoria da computação quântica. Seu algoritmo consegue, utilizando a transformada de Fourier quântica, fatorar um inteiro com uma complexidade de $O(n^2 * \log n * \log \log n)$ [Kaye 2007]. Esta complexidade é visivelmente melhor que a complexidade do GNFS e pode ser considerada polinomial como pode ser visto na figura 2.

Para exemplificar esta melhora considere uma chave RSA de 2048bits. Para quebrar esta chave precisamos fatorar-la para saber quais os dois números primos grandes foram multiplicados. Usando o algoritmo clássico teríamos que esperar cerca de 10^{29} anos. Usando o algoritmo quântico esperaríamos um total de 5 horas. Uma melhora expressiva. Outro exemplo, mais real, aconteceu em 2005 quando uma chave

RSA de 200 dígitos foi quebrada [RSALABS 2005] em um cluster com 80 computadores em aproximadamente 1 ano e 3 meses, esforço equivalente a 55 anos de processamento em um opteron 2.2Ghz. Desta forma, como a criptografia RSA é baseada na dificuldade de se fatorar um número muito grande, com os computadores quânticos as chaves RSA não guardarão mais nenhum segredo. Este problema motivou a criação da criptografia quântica.

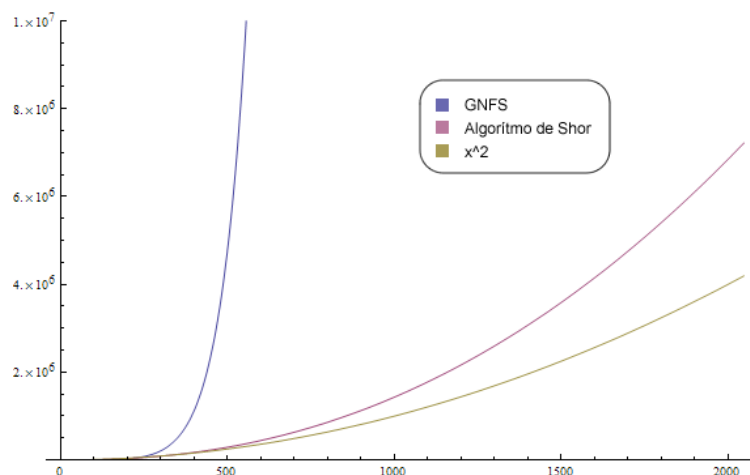


Figura 2. Comparação entre as complexidades do GNFS e o algoritmo de Shor.

O algoritmo de Shor provou portanto que os computadores quânticos podem trazer ganhos até exponenciais em tempo de execução para problemas que não têm solução eficiente sob o ponto de vista da computação clássica.

5.4. Outros Algoritmos

5.4.1. Algoritmo de Simon

Outro problema fora de BPP que obteve ganhos exponenciais com a computação quântica foi o problema de Simon: Dada uma função f como uma caixa preta e o mapeamento de f entre F_2^n e F_2^n existe um c tal que

$$f(x) = f(y) \leftrightarrow x \equiv y + c \pmod{F_2^n}$$

O algoritmo clássico para encontrar c tem complexidade $O(2^{n/2})$ e o algoritmo quântico tem complexidade $O(n^2 + nF)$ onde F é o custo de avaliar f [Shor 2002].

5.4.2. Algoritmo de Grover

Mais uma descoberta importante na área de algoritmos é o algoritmo de Grover que consegue uma complexidade de $O(\sqrt{N})$ para buscar um número em um vetor não ordenado com N itens. Sabemos que o método mais eficiente para realizar tal tarefa – a busca binária – precisa olhar para $N/2$ itens tendo, portanto, complexidade $O(N)$.

5.5. Considerações Finais Sobre Complexidade

Frente às melhoras substanciais de desempenho, algumas vezes até exponenciais, é natural questionar se $NP \subseteq BQP$, ou seja, se os computadores quânticos serão capazes de resolver de forma eficiente os problemas em NP -Completo. Suspeita-se que a resposta para esta pergunta seja não. Em [Bennet 1996] os autores mostram que uma Máquina de Turing quântica levaria tempo exponencial no tamanho da entrada para resolver problemas em NP -Completo. Vale ressaltar aqui que o próprio Shor ainda não está convencido de que a computação quântica não conseguirá lidar com problemas NP-

Completo [Shor 1996].

Como podemos ver, a teoria ainda é muito recente muitas descobertas ainda estão por vir.

Referências

Alves, F. L. “Computação Quântica: Fundamentos Físicos e Perspectivas” (2003), http://www.fisica.net/computacaoquantica/computacao_quantica_fundamentos_fisicos_e_%20perspectivas.pdf, dezembro.

Bennet, C., Bernstein, E., Brassard, G., and Vazirani, U. (1996), Strengths and Weaknesses of Quantum Computation, <http://citeseer.ist.psu.edu/bennett96strengths.html>

Kaye, P., Laflamme, R. and Mosca M. (2007), An Introduction to Quantum Computing, Oxford University Press, 1st edition.

Oliveira, Ivan S. (2002) “Computação Quântica”, <http://www.comciencia.br/reportagens/nanotecnologia/nano16.htm>, novembro.

RSALabs (2005) “RSA-200 is Factored!”, <http://www.rsa.com/rsalabs/node.asp?id=2879>

Shor, P. (1996), Polynomial-Time Algorithms for Prime Factorisation and Discrete Logarithms on a Quantum Computer, <http://arxiv.org/abs/quant-ph/9508027v2>

Shor, P. (2002) “Introduction to Quantum Algorithms”, In: Quantum Computation: A Grand Mathematical Challenge for the Twenty-First Century and the Millennium, Edited by: Samuel J. Lomonaco Jr.

Vazirani, U. (2002) ”Survey of Quantum Complexity Theory”, In: Quantum Computation: A Grand Mathematical Challenge for the Twenty-First Century and the Millennium, Edited by: Samuel J. Lomonaco Jr.