Introduction to quantum computing

Anthony Leverrier, Inria Paris
&
Mazyar Mirrahimi, Inria Paris

IQUPS course 2018

Outline of the course

courses 1 – 2: basics of quantum computing and standard algorithms (Anthony Leverrier)

- May 29 (9:15 - 10:45): basics of quantum computing: qubits, measurements, circuit model, query complexity model, Simon's algorithm
- June 5 (11:00 – 12:30): quantum Fourier transform, Shor's algorithm, Grover's algorithm

courses 3 – 4: quantum error correction and quantum fault tolerance (Mazyar Mirrahimi)

- June 18: basics of quantum error correction (discretization of errors, Shor an Steane codes) and fault-tolerance
- June 25: towards experimental implementation: surface codes and continuous-variable codes

Related material

This course is largely inspired from the remarkable set of notes by Ronald de Wolf, available online.

- Quantum Computing: Lecture Notes by Ronald de Wolf
  `http://homepages.cwi.nl/~rdewolf/qcnotes.pdf`

Other ressources include:

- the classic "Quantum computation and quantum information" by Nielsen & Chuang
- Lecture notes by John Preskill
  `http://www.theory.caltech.edu/people/preskill/ph229/`

The end of Moore's law

Intel Delays Mass Production of 10 nm CPUs to 2019

by Anton Shilov on April 27, 2018 12:20 PM EST

`https://www.anandtech.com/show/12693/`

`intel-delays-mass-production-of-10-nm-cpus-to-2019`

	Intel First Production
1999	180 nm
2001	130 nm
2003	90 nm
2005	65 nm
2007	45 nm
2009	32 nm
2011	22 nm
2014	14 nm
2016	~~10 nm~~
2017	~~10 nm~~
2018	10 nm?
2019	10 nm!

Why study quantum computing?

quantum computation

- investigation of the computational power of computer based on quantum mechanical principles
- main objective: find algorithms with speedup compared to classical algos

Motivations

- *miniaturization* reaches levels where quantum effects become non-negligible. One can either try to suppress them or to exploit them.
- *speedups* for computation, but also applications in cryptography
- objective is to understand the power of the strongest-possible computing devices allowed by *Nature*

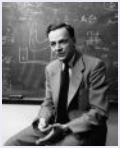
Genesis of quantum computing

Feynman 1981

"Can quantum systems be probabilistically simulated by a classical computer? [...] The answer is almost certainly, No!"

$\Longrightarrow$ use quantum systems to simulate quantum systems!

$\Longrightarrow$ birth of quantum simulation



Deutsch 1985

- quantum Turing machine
- existence of a universal machine

$\Longrightarrow$ birth of quantum computing

Bernstein, Vazirani 1993

- efficient quantum Turing machine (complexity class BQP)
- Bernstein-Vazirani problem: $f : \{0,1\}^n \to \{0,1\}$ such that $f(x) = a \cdot x$
  Find a. $\Longrightarrow$ ok with 1 quantum query vs n classically

The first algorithms

Simon, Shor 1994

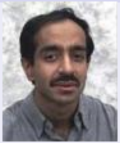
exponential speedups for

- period finding
- factoring!! *very surprising* $\Longrightarrow$ sparked a lot of interest in the field
- discrete logarithm

$\Longrightarrow$ exploits Quantum Fourier Transform

$\Longrightarrow$ consequences for public-key cryptography: breaks most cryptosystems deployed today

Grover 1996



- search an n-item list with $O(\sqrt{n})$ queries
- lots of applications (find collisions, approximate counting, shortest path)

but only quadratic improvement

Basics of quantum computation

States, evolution, measurement

- in this course, we restrict ourselves to *pure n-qubit states*: $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$

$$|\psi\rangle = \alpha_{0\dots00}|0\cdots00\rangle + \alpha_{0\dots01}|0\cdots01\rangle\cdots + \alpha_{1\dots11}|1\cdots11\rangle$$

$$\text{with} \quad \sum |\alpha_{\vec{i}}|^2 = 1 \quad \text{(normalization)} \quad \text{and} \quad |i_1 i_2 \cdots i_n\rangle := |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle$$

  in practice, one needs to deal with *decoherence*, and therefore *mixed states* but quantum fault-tolerance techniques can be applied to deal with such issues (threshold theorem): see Mazyar's course
- the state is *evolved unitarily*, possibly by applying the unitary U (such that $UU^\dagger = \mathbb{1}$) also on ancilla qubits initialized in $|0\rangle^{\otimes m}$:

$$|\psi\rangle \mapsto U|\psi\rangle|0\rangle^{\otimes m}$$

- in this course, states are *measured in the computational (standard) basis*: the measurement returns the string $\vec{i} \in \{0,1\}^n$ with probability

$$\mathbb{P}(\vec{i}) = |\langle \vec{i}|\psi\rangle|^2 = |\alpha_{\vec{i}}|^2$$

Elementary gates

gate: unitary acting on a small number of qubits (typically between 1 and 3), similar to classical logic gates AND, OR and NOT

single-qubit gates

- bitflip gate X: $|0\rangle \leftrightarrow |1\rangle$ $\qquad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
- phase-flip gate Z: $|0\rangle \mapsto |0\rangle, \quad |1\rangle \mapsto -|1\rangle$ $\qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
- phase-flip gate R_ϕ: $|0\rangle \mapsto |0\rangle, \quad |1\rangle \mapsto e^{i\phi}|1\rangle$ $\qquad R_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$ $\qquad T := R_{\pi/4}$
- Hadamard gate: $|0\rangle \leftrightarrow |+\rangle, \qquad |1\rangle \leftrightarrow |-\rangle$ $\qquad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

$$|\pm\rangle := \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$$

Elementary gates

two-qubit gates

- controlled-not (CNOT): flips the second input qubit if the first one is $|1\rangle$, and does nothing if the first qubit is $|0\rangle$

$$\begin{aligned}\mathrm{CNOT}|0\rangle|b\rangle &= |0\rangle|b\rangle \\ \mathrm{CNOT}|1\rangle|b\rangle &= |1\rangle|1-b\rangle\end{aligned}$$

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- controlled-U (for single-qubit unitary U):

$$\begin{pmatrix} \mathbb{1}_2 & 0 \\ 0 & U \end{pmatrix}$$

Models of quantum computing

Models of quantum computing

There are different models to describe how a quantum computer can apply computational steps to its registers of qubits.

- *quantum Turing machine* (Deutsch 1985: states, tape, transition function...)
- *circuit model:* this course
- *adiabatic quantum computing:*
  - encode your problem as a Hamiltonian H and the solution as a ground state
  - start with a ground state of an easy Hamiltonian H_0
  - slowly evolve the system by applying $(1-\alpha(t))H_0 + \alpha(t)H$ for $\alpha(t_{\mathrm{init}}) = 0, \alpha(t_{\mathrm{fin}}) = 1$
  - provided that the evolution is sufficiently slow, one remains in the ground state
- *measurement-based quantum computing* (Raussendorf, Briegel 2002):
  - start with a generic highly entangled state: a *cluster state*
  - measure each qubit one by one and update following measurement angles as a function of previous measurement results

Theorem

These models are equivalent: they can simulate each other in polynomial time

The circuit model

We are mostly interested in classical problems where the input is some n-bit string $x \in \{0,1\}^n$, and we want an output $y \in \{0,1\}^m$, possibly with $m = 1$.

- input state: $|\vec{x}\rangle \otimes |0\rangle^{\otimes n'}$ (input + ancilla)
- unitary operation: U described as a quantum network of elementary gates
- output: measure the final $(n+n')$-qubit state in the computational basis

Note that the answer is generally probabilistic. Sometimes we repeat the process a few times and take a majority vote.

Question

can any unitary operation U acting on N qubits be decomposed into a circuit of elementary gates acting on 1 or 2 qubits?

$\Longrightarrow$ universal gate set: reduces to infinitely-many elementary gates

$\Longrightarrow$ Kitaev-Solovay theorem: approximate unitary with finite gate set

Universality of simple gate sets

universal gate set

Any unitary on N qubits can be decomposed using

- arbitrary single qubit gates
- the 2-qubit CNOT gate

Problem: it is not realistic to be able to perform arbitrary single-qubit gates with infinite precision. We would like a finite gate set.

Kitaev-Solovay theorem

The following sets allow to approximate any unitary arbitrarily well:

- CNOT, Hadamard H, T-gate $\mathrm{T} = \mathrm{R}_{\pi/4}$
- Hadamard and Toffoli (3-qubit gate CCNOT) if the unitary have only real entries

Solovay-Kitaev: any 1 or 2-qubit unitary can be approximated up to error ε using $\mathrm{polylog}(1/\varepsilon)$ gates from the set.

Quantum parallelism

The main motivation for quantum computation: "perform many computations in superposition".

Lemma

Suppose we have a classical algorithm that computes some function $f : \{0,1\}^n \to \{0,1\}^m$. Then we can build a quantum circuit U_f consisting only of Toffoli gates that maps

$$U_f \quad : \quad |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle.$$

Not $|x\rangle \mapsto |f(x)\rangle$ *…not unitary in general!*

Consequence:

$$H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

$$U\left(\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|0\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|f(x)\rangle$$

Quantum parallelism

The main motivation for quantum computation: "perform many computations in superposition".

Lemma

Suppose we have a classical algorithm that computes some function $f : \{0,1\}^n \to \{0,1\}^m$. Then we can build a quantum circuit U_f consisting only of Toffoli gates that maps

$$U_f \quad : \quad |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle.$$

Caution!

- One applies U_f just once, but the final state contains $f(x)$ for all 2^n input values.
- However, measuring the output state in the computational basis only yields a single (random) couple $(x, f(x))$.
- *Holevo theorem:* one cannot extract more than n bits of information from n qubits

The early quantum algorithms

Query complexity model

Standard circuit model: input of computation is encoded in the input state; quantum circuit; measurement in computational basis ... *how many gates?*

Query complexity model: the input (e.g. a function) is accessed as a *black box*

N-*bit input* $x = (x_1, \cdots, x_N) \in \{0,1\}^N$

- Usually, $N = 2^n$: bit x_i can be addressed with n-bit string i.
- Example: x is a *Boolean function* $f : \{0,1\}^n \to \{0,1\}, \quad f(i) \equiv x_i$
- input = N-bit memory (Random Access Memory) which can be accessed as a black-box at any point we want.
- modeled as a quantum unitary on $n+1$ qubits (n-bit address and single-bit target)

$$O_x : |i, 0\rangle \mapsto |i, x_i\rangle$$

$$O_x : |i, b\rangle \mapsto |i, x_i \oplus b\rangle$$

- alternative phase-oracle: $O_{x,\pm} : |i\rangle \mapsto (-1)^{x_i}|i\rangle$

Some early algorithms

provide speedups in query complexity model, not in the standard circuit model

Deutsch-Jozsa (1992)

For $N = 2^n$, we are given $x \in \{0,1\}^N$ either

- constant: all x_i are equal
- balanced: half of x_i are 0, half are 1

Find which one.

Bernstein-Vazirani (1993)

For $N = 2^n$, we are given $x \in \{0,1\}^N$ such that $\exists a \in \{0,1\}^n$ with $x_i = (i \cdot a) \mod 2$.
Find a.

Simon (1994)

For $N = 2^n$, we are given $x = (x_1, \cdots, x_N)$ with $x_i \in \{0,1\}^n$ with the property that $\exists s \neq 0 \in \{0,1\}^n$ such that $\quad x_i = x_j \iff (i = j \quad \text{or} \quad i = j \oplus s)$.
Find s.

Deutsch-Josza

the problem

For $N = 2^n$, we are given $x \in \{0,1\}^N$ either

- constant: all x_i are equal
- balanced: half of x_i are 0, half are 1

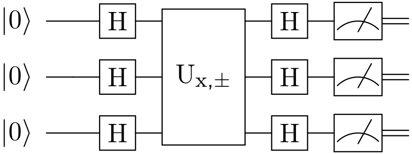
Find which one.

complexity

- classical deterministic (no errors): at least $N/2 + 1$ queries needed
- classical if errors are allowed: constant number of queries
- quantum: single query!

$\Longrightarrow$ separation *quantum* vs *exact classical*

Deutsch-Josza



$$|0^n\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \sum_{i\in\{0,1\}^n} |i\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \sum_{i\in\{0,1\}^n} (-1)^{x_i} |i\rangle$$
$$\longrightarrow \frac{1}{\sqrt{2^n}} \sum_{i\in\{0,1\}^n} (-1)^{x_i} \sum_{j\in\{0,1\}^n} (-1)^{i\cdot j} |j\rangle$$

Amplitude of $|0^n\rangle$ state:

$$\frac{1}{\sqrt{2^n}} \sum_{i\in\{0,1\}^n} (-1)^{x_i} = \begin{cases} 1 & \text{if } x_i = 0 \quad \forall i \\ \text{-}1 & \text{if } x_i = 1 \quad \forall i \\ 0 & \text{if } x \text{ is balanced} \end{cases}$$

Yields $|0^n\rangle$ iff x is constant: 1 query and $O(n)$ operations

Bernstein-Vazirani

the problem: linear function, find coefficients

For $N = 2^n$, we are given $x \in \{0,1\}^N$ such that $\exists a \in \{0,1\}^n$ with $x_i = (i \cdot a) \mod 2$.
Find a.

complexity

- randomized classical, small errors allowed: needs at least n queries (each query gives at most 1 bit of info)
- quantum: single query!

same algorithm as Deusch-Josza: $(-1)^{x_i} = (-1)^{(i \cdot a) \mod 2} = (-1)^{i \cdot a}$
state after the query:

$$\frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} (-1)^{x_i} |i\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} (-1)^{i \cdot a} |i\rangle$$

$H = H^{-1} \implies |a\rangle$

Simon's algorithm

Exponential speedup for query complexity (we count queries, not ordinary operations)

hidden period for 2-to-1 function

For $N = 2^n$, we are given $x = (x_1, \cdots, x_N)$ with $x_i \in \{0,1\}^n$ with the property that $\exists s \neq 0 \in \{0,1\}^n$ such that

$$x_i = x_j \iff (i = j \quad \text{or} \quad i = j \oplus s).$$

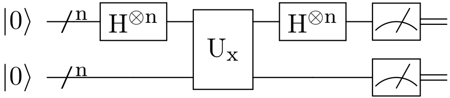
Find s.

Note that x_i is an n-bit string, not a single bit.

complexity

- randomized classical algorithm in $O(\sqrt{2^n})$ queries with birthday paradox
- this is essentially optimal for classical algorithms
- quantum (Simon's algorithm): $O(n)$ queries

$\Longrightarrow$ exponential separation *quantum* vs *randomized classical*



$$|0^n\rangle|0^n\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \sum_{i\in\{0,1\}^n} |i\rangle|0^n\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \sum_{i\in\{0,1\}^n} |i\rangle|x_i\rangle$$

Measure 2nd n-bit register: yields $x_i \in \{0,1\}^n$, collapses the first register to superposition of 2 indices compatible with x_i

$$\frac{1}{\sqrt{2}}(|i\rangle + |i \oplus s\rangle)|x_i\rangle$$

Hadamard to first n qubits:

$$\frac{1}{\sqrt{2^{n+1}}}\left(\sum_{j\in\{0,1\}^n}(-1)^{i\cdot j}|j\rangle + \sum_{j\in\{0,1\}^n}(-1)^{(i\oplus s)\cdot j}|j\rangle\right) = \frac{1}{\sqrt{2^{n+1}}}\sum_{j\in\{0,1\}^n}(-1)^{i\cdot j}(1+(-1)^{s\cdot j})|j\rangle$$

Simon's algorithm

Measure state

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{j\in\{0,1\}^n} (-1)^{i\cdot j}(1+(-1)^{s\cdot j})|j\rangle$$

- $|j\rangle$ has nonzero amplitude iff $s \cdot j = 0 \mod 2$.
- The measurement outcome is uniformly drawn from $\{j \,|\, s \cdot j = 0 \mod 2\}$.
- $\implies$ linear equation giving information about s
- repeat until we get $n - 1$ independent linear equations
- solutions are 0 and s via Gaussian elimination (classical circuit of size $O(n^3)$)

$\implies$ exponential speedup in the query complexity model! Can we get it in the standard model as well?

Recap

- quantum computers can exploit quantum parallelism, but cannot really do an exponential number of computations in parallel
- one single output!
- different models of quantum computing: circuit, measurement-based, adiabatic computing, all equivalent (up to polynomials)

Today: "speedup" in query complexity model

- black-box access to a function
- provable, exponential improvement, but not in a real situation

Next week: speedup in standard gate complexity model

- Shor's algorithm for factoring
- Grover's algorithm for search