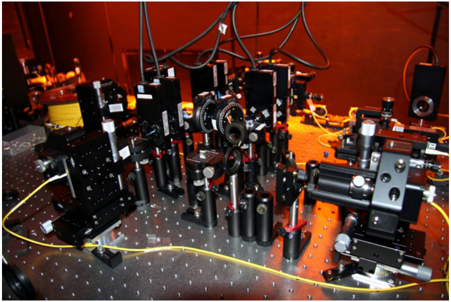
# Introduction to Quantum Computing
## Part I

Emma Strubell

http://cs.umaine.edu/~ema/quantum_tutorial.pdf

April 12, 2011



# Overview

What is quantum computing?

- Background
- Caveats

Mathematical representation

- Fundamental differences
- Hilbert spaces and Dirac notation
- The qubit
- Quantum Registers
- Quantum logic gates
- Computational complexity

# Outline

What is quantum computing?
- Background
- Caveats

Mathematical representation
- Fundamental differences
- Hilbert spaces and Dirac notation
- The qubit
- Quantum Registers
- Quantum logic gates
- Computational complexity

# Origins of fame

- Quantum computer first proposed by Richard Feynman in 1981
  - Problem: efficiently simulating quantum systems inherently impossible on a classical computer
  - Solution: new machine "built of quantum mechanical elements which obey quantum mechanical laws"
- Daniel Simon demonstrates exponential speedup in 1994
  - nobody cares; algorithm too abstract
- Peter Shor demonstrates *exciting* exponential speedup in 1997
  - based on Simon's algorithm
  - efficiently factors integers into primes
  - this breaks RSA


# Outline

What is quantum computing?

- Background
- Caveats

Mathematical representation

- Fundamental differences
- Hilbert spaces and Dirac notation
- The qubit
- Quantum Registers
- Quantum logic gates
- Computational complexity

# Unfortunately, scalable QCs still don't exist

- As of 2009, quantum computers able to factor 15 into 5 and 3
- The problem is *decoherence*
  - Man-made quantum system wants to interact with surrounding systems
  - Sources of interference include electric and magnetic fields required to power machine itself

# Overview

What is quantum computing?
- Background
- Caveats

Mathematical representation
- Fundamental differences
- Hilbert spaces and Dirac notation
- The qubit
- Quantum Registers
- Quantum logic gates
- Computational complexity

# Outline

What is quantum computing?

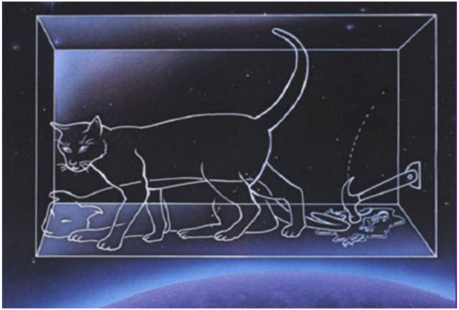
- Background
- Caveats

Mathematical representation

- Fundamental differences
- Hilbert spaces and Dirac notation
- The qubit
- Quantum Registers
- Quantum logic gates
- Computational complexity

# Three main differences from classical computers

1. Superposition
   - quantum system exists in all possible states at all times
2. Probabilities
   - fortunately, a probability can be associated with each of those states
3. Entanglement
   - probabilities of different states can depend on each other
   - quantum teleportation uses this property for cryptographic purposes



# Outline

# Dirac notation

- Just another way of describing vectors:

$$\mathbf{v} = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_n \end{bmatrix} = |\mathbf{v}\rangle$$

- and their duals:

$$\langle \mathbf{v}| = \overline{\mathbf{v}^{\mathrm{T}}} = \begin{bmatrix} \overline{v_0} & \overline{v_1} & \dots & \overline{v_n} \end{bmatrix}$$

- Convenient for describing vectors in the Hilbert space $\mathbb{C}^n$, the vector space of quantum mechanics

# $\mathbb{C}^n$ and the inner product

- A Hilbert space, for our (finite) purposes, is a vector space with an *inner product*, and a *norm* defined by that inner product. We use the following in $\mathbb{C}^n$:
  - The inner product assigns a scalar value to each pair of vectors:

$$\langle \mathbf{u}|\mathbf{v}\rangle = \overline{\mathbf{u}^{\mathrm{T}}}\mathbf{v} = \begin{bmatrix} \overline{u_0} & \overline{u_1} & \dots & \overline{u_n} \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_n \end{bmatrix} = \overline{u_0}\cdot v_0 + \overline{u_1}\cdot v_1 + \ldots + \overline{u_n}\cdot v_n$$

  - The norm is the square root of the inner product of a vector with itself (i.e. Euclidean norm, $\ell^2$-norm, 2-norm over complex numbers):

$$\| |\mathbf{v}\rangle \| = \sqrt{\langle \mathbf{v}|\mathbf{v}\rangle}$$

  - Geometrically, this norm gives the distance from the origin to the point $|\mathbf{v}\rangle$ that follows from the Pythagorean theorem.

# Properties of the inner product

The inner product satisfies the three following properties:

**Definition**

1. $\langle \mathbf{v}|\mathbf{v}\rangle \geq 0$, with $\langle \mathbf{v}|\mathbf{v}\rangle = 0$ if and only if $|\mathbf{v}\rangle = \mathbf{0}$.
2. $\langle \mathbf{u}|\mathbf{v}\rangle = \overline{\langle \mathbf{v}|\mathbf{u}\rangle}$ for all $|\mathbf{u}\rangle$, $|\mathbf{v}\rangle$ in the vector space.
3. $\langle \mathbf{u}|\alpha_0\mathbf{v} + \alpha_1\mathbf{w}\rangle = \alpha_0 \langle \mathbf{u}|\mathbf{v}\rangle + \alpha_1 \langle \mathbf{u}|\mathbf{w}\rangle$.
   More generally, the inner product of $|\mathbf{u}\rangle$ and $\sum_i \alpha_i |\mathbf{v}_i\rangle$ is equal to $\sum_i \alpha_i \langle \mathbf{u}|\mathbf{v}_i\rangle$ for all scalars $\alpha_i$ and vectors $|\mathbf{u}\rangle$, $|\mathbf{v}\rangle$ in the vector space (this is known as *linearity in the second argument*).

# The outer product

- The *outer product* is the *tensor* or *Kronecker product* of a vector with the conjugate transpose of another. The result is not a scalar, but a matrix:

$$|\mathbf{v}\rangle\langle\mathbf{u}| = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_n \end{bmatrix} \begin{bmatrix} \overline{u_0} & \overline{u_1} & \dots & \overline{u_m} \end{bmatrix} = \begin{bmatrix} v_0\overline{u_0} & v_0\overline{u_1} & \dots & v_0\overline{u_m} \\ v_1\overline{u_0} & v_1\overline{u_1} & \dots & v_1\overline{u_m} \\ \vdots & \vdots & \ddots & \vdots \\ v_n\overline{u_0} & v_n\overline{u_1} & \dots & v_n\overline{u_m} \end{bmatrix}$$

- Often used to describe a linear transformation between vector spaces.
- A linear transformation from a Hilbert space $U$ to another Hilbert space $V$ on a vector $|\mathbf{w}\rangle$ in $U$ may be succintly described in Dirac notation:

$$(|\mathbf{v}\rangle\langle\mathbf{u}|)|\mathbf{w}\rangle = |\mathbf{v}\rangle\langle\mathbf{u}|\mathbf{w}\rangle = \langle\mathbf{u}|\mathbf{w}\rangle|\mathbf{v}\rangle$$

  Since $\langle\mathbf{u}|\mathbf{w}\rangle$ is a commutative, scalar value.

# The tensor product

- Usually simplified from $|\mathbf{u}\rangle \otimes |\mathbf{v}\rangle$ to $|\mathbf{u}\rangle\,|\mathbf{v}\rangle$ or $|\mathbf{uv}\rangle$
- A vector tensored with itself $n$ times is denoted $|\mathbf{v}\rangle^{\otimes n}$ or $|\mathbf{v}\rangle^{n}$
- Two column vectors $|\mathbf{u}\rangle$ and $|\mathbf{v}\rangle$ of lengths $m$ and $n$ yield a column vector of length $m \cdot n$ when tensored:

$$|\mathbf{u}\rangle\,|\mathbf{v}\rangle = |\mathbf{uv}\rangle = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_m \end{bmatrix} \otimes \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} u_0 \cdot v_0 \\ u_0 \cdot v_1 \\ \vdots \\ u_0 \cdot v_n \\ u_1 \cdot v_0 \\ \vdots \\ u_{m-1} \cdot v_n \\ u_m \cdot v_0 \\ \vdots \\ u_m \cdot v_n \end{bmatrix}$$

# Outline

# $\mathbb{C}^2$ describes a single quantum bit (qubit)

- A classical bit may be represented as a base-2 number that takes either the value 1 or the value 0
- Qubits are also base-2 numbers, but in a superposition of the measurable values 1 and 0
- The state of a qubit at any given time represented as a two-dimensional *state space* in $\mathbb{C}^2$ with *orthonormal* basis vectors $|1\rangle$ and $|0\rangle$
- The superposition $|\psi\rangle$ of a qubit is represented as a linear combination of those basis vectors:

$$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle$$

Where $a_0$ is the complex scalar *amplitude* of measuring $|0\rangle$, and $a_1$ the amplitude of measuring the value $|1\rangle$.

# Amplitudes, not probabilities

- Amplitudes may be thought of as "quantum probabilities" in that they represent the chance that a given quantum state will be observed when the superposition is collapsed
- Most fundamental difference between probabilities of states in classical probabilistic algorithms and amplitudes: amplitudes are complex
  - Complex numbers required to fully describe superposition of states, interference or entanglement in quantum systems.[1]
  - As the probabilities of a classical system must sum to 1, so too the squares of the absolute values of the amplitudes of states in a quantum system must add up to 1

[1] See `http://www.scottaaronson.com/democritus/lec9.html` for a great discussion by of why complex numbers and the 2-norm are used to describe quantum mechanical systems

# Amplitudes and the normalization condition

- Just as the hardware underlying the bits of a classical computer may vary in voltage, quantum systems are not usually so perfectly behaved
- An assumption is made about quantum state vectors called the *normalization conditon*: $|\psi\rangle$ is a unit vector.
  - $\||\psi\rangle\| = \langle\psi|\psi\rangle = 1$
  - If $|0\rangle$ and $|1\rangle$ are orthonormal, then by orthogonality $\langle 0|1\rangle = \langle 1|0\rangle = 0$, and by normality $\langle 0|0\rangle = \langle 1|1\rangle = 1$
  - It follows that $|a_0|^2 + |a_1|^2 = 1$:

$$\begin{aligned} 1 &= \langle\psi|\psi\rangle \\ &= (\overline{a_0}\langle 0| + \overline{a_1}\langle 1|) \cdot (a_0|0\rangle + a_1|1\rangle) \\ &= |a_0|^2\langle 0|0\rangle + |a_1|^2\langle 1|1\rangle + \overline{a_1}a_0\langle 1|0\rangle + \overline{a_0}a_1\langle 0|1\rangle \\ &= |a_0|^2 + |a_1|^2 \end{aligned}$$

# Why we use Dirac notation

The following is equivalent to the last slide:

$$
\begin{aligned}
1 &= \langle\psi|\psi\rangle \\
&= \left(\overline{a_0}\,\langle 0| + \overline{a_1}\,\langle 1|\right)\cdot\left(a_0\,|0\rangle + a_1\,|1\rangle\right) \\
&= \left(\overline{a_0}\begin{bmatrix}\overline{\psi_{00}} & \overline{\psi_{01}}\end{bmatrix} + \overline{a_1}\begin{bmatrix}\overline{\psi_{10}} & \overline{\psi_{11}}\end{bmatrix}\right)\cdot\left(a_0\begin{bmatrix}\psi_{00}\\ \psi_{01}\end{bmatrix} + a_1\begin{bmatrix}\psi_{10}\\ \psi_{11}\end{bmatrix}\right) \\
&= \begin{bmatrix}\overline{a_0\psi_{00}} + \overline{a_1\psi_{10}} & \overline{a_0\psi_{01}} + \overline{a_1\psi_{11}}\end{bmatrix}\cdot\begin{bmatrix}a_0\psi_{00} + a_1\psi_{10}\\ a_0\psi_{01} + a_1\psi_{11}\end{bmatrix} \\
&= \overline{a_0\psi_{00}}a_0\psi_{00} + \overline{a_1\psi_{10}}a_0\psi_{00} + \overline{a_0\psi_{00}}a_1\psi_{10} + \overline{a_1\psi_{10}}a_1\psi_{10} \\
&\quad + \overline{a_0\psi_{01}}a_0\psi_{01} + \overline{a_1\psi_{11}}a_0\psi_{01} + \overline{a_0\psi_{01}}a_1\psi_{11} + \overline{a_1\psi_{11}}a_1\psi_{11} \\
&= |a_0|^2\left(|\psi_{00}|^2 + |\psi_{01}|^2\right) + |a_1|^2\left(|\psi_{10}|^2 + |\psi_{11}|^2\right) \\
&\quad + \overline{a_1}a_0\left(\overline{\psi_{10}}\psi_{00} + \overline{\psi_{11}}\psi_{01}\right) + \overline{a_0}a_1\left(\overline{\psi_{00}}\psi_{10} + \overline{\psi_{01}}\psi_{11}\right) \\
&= |a_0|^2 + |a_1|^2
\end{aligned}
$$

# The computational basis

- $|0\rangle$ and $|1\rangle$ may be transformed into any two vectors that form an orthonormal basis in $\mathbb{C}^2$
- The most common basis used in quantum computing is called the *computational basis*:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- The computational basis tends to be the most straightforward basis for computing and understanding quantum algorithms
- Assume I'm using the computational basis unless otherwise stated

# Another basis

- Any other orthonormal basis could be used:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$$

- Providing a slightly different but equivalent way of expressing of a qubit:

$$\begin{aligned} |\psi\rangle &= a_0 |0\rangle + a_1 |1\rangle \\ &= a_0 \frac{|+\rangle + |-\rangle}{\sqrt{2}} + a_1 \frac{|+\rangle - |-\rangle}{\sqrt{2}} \\ &= \frac{a_0 + a_1}{\sqrt{2}} |+\rangle + \frac{a_0 + a_1}{\sqrt{2}} |-\rangle \end{aligned}$$

- Here, instead of measuring the states $|0\rangle$ and $|1\rangle$ each with respective probabilities $|a_0|^2$ and $|a_1|^2$, the states $|+\rangle$ and $|-\rangle$ would be measured with probabilities $|a_0 + a_1|^2/2$ and $|a_0 - a_1|^2/2$.

# Outline

# Registers more useful than single qubits

- Each qubit in a quantum register is in a superposition of $|1\rangle$ and $|0\rangle$
- Consequently, a register of $n$ qubits is in a superposition of all $2^n$ possible bit strings that could be represented using $n$ bits
- The state space of a size-$n$ quantum register is a linear combination of $n$ basis vectors, each of length $2^n$:

$$|\psi_n\rangle = \sum_{i=0}^{2^n-1} a_i |i\rangle$$

- A three-qubit register would thus have the following expansion:

$$\begin{aligned} |\psi_2\rangle = a_0 |000\rangle + a_1 |001\rangle + a_2 |010\rangle + a_3 |011\rangle \\ + a_4 |100\rangle + a_5 |101\rangle + a_6 |110\rangle + a_7 |111\rangle \end{aligned}$$

# Registers continued

- Each possible bit configuration in the quantum superposition is denoted by the tensor product of its counterpart qubits
- Consider $|101\rangle$, the bit string that represents the integer value 5:

$$\begin{aligned} |101\rangle &= |1\rangle \otimes |0\rangle \otimes |1\rangle \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}} \end{aligned}$$

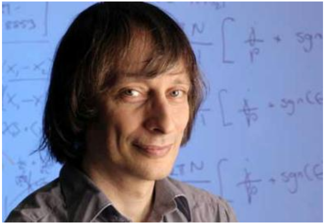
- As with single qubits, the squared absolute value of the amplitude associated with a given bit string is the probability of observing that bit string, and the the sqares of the absolute values of the amplitudes of all $2^n$ possible bit configuations of an $n$-bit register sum to unity:

$$\sum_{i=0}^{2^n-1} |a_i|^2 = 1$$

# Outline

# Evolving the system: quantum circuits and quantum gates

- One way of thinking about algorithm design and computation is via quantum Turing machines
- First described by David Deutsch in 1985, but both a quantum Turing machine's tape and its read-write head exist in superpositions of an exponential number states!



- Instead of using the Turing machine as a computational model, operations on a quantum computer most often described using quantum circuits (also introduced by Deutsch a few years later)
- Although circuits are computationally equivalent to Turing machines, they are usually much simpler to depict, manipulate and understand

# Quantum gates represent unitary transformations

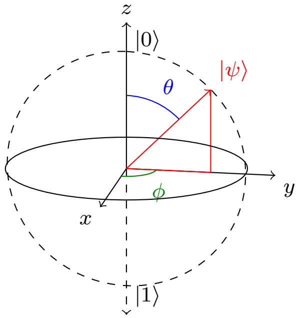
- Quantum gates are represented as transformation matrices, linear operators applied to a quantum register by tensoring the operator with the register
- All quantum linear operators must be *unitary*:
  - If a complex matrix $U$ is unitary, then $U^{-1} = U^{\dagger}$, where $U^{\dagger}$ is the conjugate transpose: $U^{\dagger} = \overline{U}^{\mathrm{T}}$
  - It follows that $UU^{\dagger} = U^{\dagger}U = I$
  - Unitary operators preserve inner product:

$$\langle \mathbf{u} | U^{\dagger}U | \mathbf{v} \rangle = \langle \mathbf{u} | I | \mathbf{v} \rangle = \langle \mathbf{u} | \mathbf{v} \rangle$$

  - The composition of two unitary operators is also unitary:

$$(UV)^{\dagger} = V^{\dagger}U^{\dagger} = V^{-1}U^{-1} = (UV)^{-1}$$

# The Bloch sphere




- Unitary transformations performed on a qubit may be visualized as rotations and reflections about the $x$, $y$, and $z$ axes of the *Bloch sphere*
- All linear combinations $a_0 |0\rangle + a_1 |1\rangle$ in $\mathbb{C}^2$ correspond to all the points $(\theta, \psi)$ on the surface of the unit sphere, where $a_0 = \cos(\theta/2)$ and $a_1 = e^{i\phi} \sin(\theta/2) = (\cos\phi + i \sin\phi) \sin\frac{\theta}{2}$

# The Hadamard operator

$$-\boxed{H}- = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}\langle 0| + \frac{|0\rangle - |1\rangle}{\sqrt{2}}\langle 1|$$

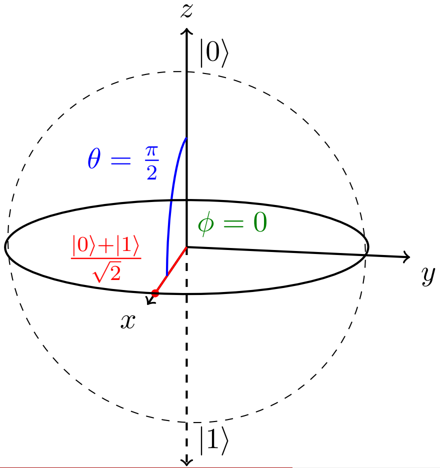
- Often referred to as a "fair coin flip," the Hadamard operator applied to a qubit with the value $|0\rangle$ or $|1\rangle$ will induce an equal superposition of the states $|0\rangle$ and $|1\rangle$:

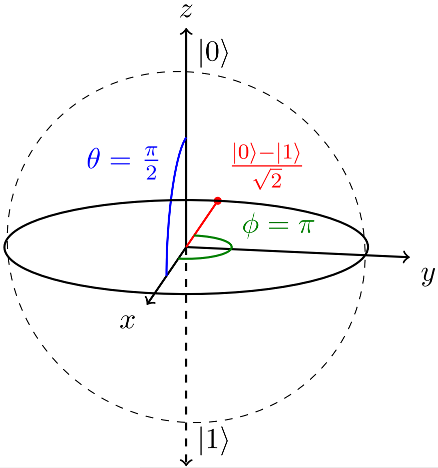
$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}\langle 0|0\rangle + \frac{|0\rangle - |1\rangle}{\sqrt{2}}\langle 1|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$
$$H|1\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}\langle 0|1\rangle + \frac{|0\rangle - |1\rangle}{\sqrt{2}}\langle 1|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

- Many quantum algorithms begin by applying the Hadamard operator to each qubit in a register initialized to $|0\rangle^n$, which puts the entire register into an equal superposition of states

# Bloch sphere representation of the Hadamard operator

- Geometrically, the Hadamard operator performs a rotation of $\pi/2$ about the $y$ axis followed by a rotation about the $x$ axis by $\pi$ radians on the Bloch sphere:



# The Pauli gates

- The three Pauli gates, named after yet another Nobel laureate Wolfgang Pauli, are also important single-qubit gates for quantum computation
- The Pauli-X gate swaps the amplitudes of $|0\rangle$ and $|1\rangle$:

$$-\boxed{X}- = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = |1\rangle\langle 0| + |0\rangle\langle 1|$$

- The Pauli-Y gate swaps the amplitudes of $|0\rangle$ and $|1\rangle$, multiplies each amplitude by $i$, and negates the amplitude of $|1\rangle$:

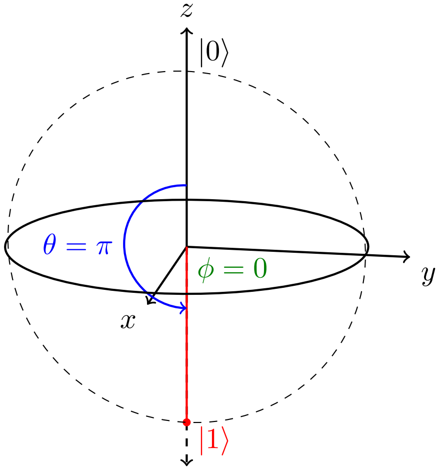
$$-\boxed{Y}- = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = i\,|1\rangle\langle 0| - i\,|0\rangle\langle 1|$$

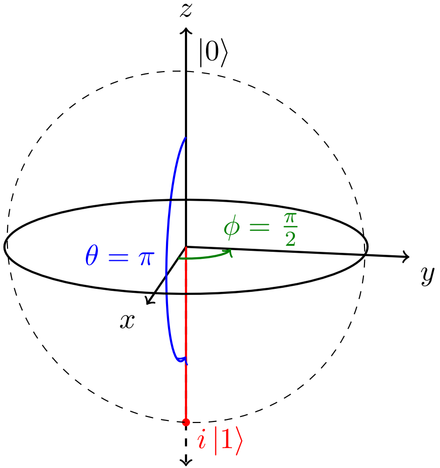
- And the Pauli-Z gate negates the amplitude of $|1\rangle$, leaving the amplitude of $|0\rangle$ the same:

$$-\boxed{Z}- = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = |1\rangle\langle 0| - |0\rangle\langle 1|$$

# Bloch sphere representation of Pauli-X and -Y gates

- The Pauli-X, -Y, and -Z gates correspond to rotations by $\pi$ radians about the $x$, $y$, and $z$ axes respectively on the Bloch sphere



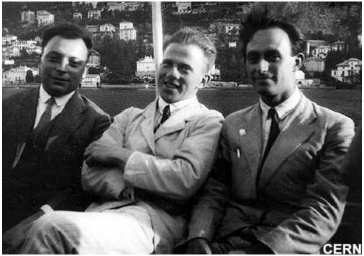



# Generalized phase shift

- The Pauli-Z gate, altering only the phase of the system, is a special case of the more general phase-shift gate, which does not modify the amplitude of $|0\rangle$ but changes the phase of $|1\rangle$ by a factor of $e^{i\theta}$ for any value of $\theta$:

$$-\boxed{R_\theta}- = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} = |1\rangle\langle 0| + e^{i\theta}|0\rangle\langle 1|$$

- The Pauli-Z gate is equivalent to the phase-shift gate with $\theta = \pi$.
- Wolfgang Pauli with friends Werner Heisenberg and Enrico Fermi:



# More phase shift gates

- Another special case of the phase-shift gate where $\theta = \pi/2$ is known as simply the phase gate, denoted $S$, which changes the phase of $|1\rangle$ by a factor of $i$:

$$
-\boxed{S}- = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} = |1\rangle\langle 0| + i\,|0\rangle\langle 1|
$$

- And the phase-shift gate where $\theta = \pi/4$ is referred to as the $\pi/8$ gate, or $T$:

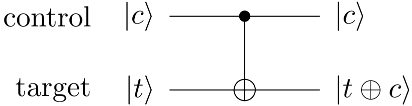
$$
-\boxed{T}- = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} = |1\rangle\langle 0| + e^{i\pi/4}\,|0\rangle\langle 1|
$$

With the name $\pi/8$ coming from the fact that this transformation can also be written as a matrix with $\pi/8$ along the diagonal:

$$
\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} = e^{i\pi/8} \begin{bmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{bmatrix}
$$

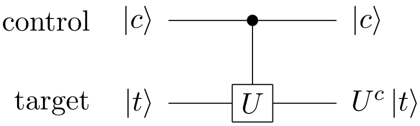
# Controlled operations: CNOT

- Quantum computing also makes use of *controlled operations*, multi-qubit operations that change the state of a qubit based on the values of other qubits
- The quantum controlled-NOT or CNOT gate swaps the amplitudes of the $|0\rangle$ and $|1\rangle$ basis states of a qubit, equivalent to application of the Pauli-X gate, only if the controlling qubit has the value $|1\rangle$:



# Generalized controlled operations

- Controlled operations are not restricted to conditional application of the Pauli-X gate; Any unitary operation may be performed:




- Matrix representation:

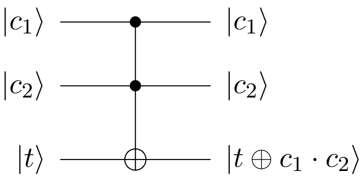
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & x_{00} & x_{10} \\ 0 & 0 & x_{01} & x_{11} \end{bmatrix}$$

- Dirac equivalent:

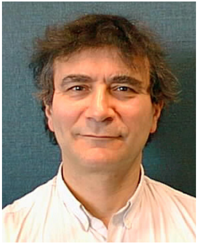
$$|00\rangle\langle 00| + |01\rangle\langle 01| + x_{00}\,|10\rangle\langle 10| + x_{01}\,|10\rangle\langle 11| \\ + x_{10}\,|11\rangle\langle 10| + x_{11}\,|11\rangle\langle 11|$$

# Controlled operations: Toffoli

- In fact, controlled operations are possible with any number $n$ control qubits and any unitary operator on $k$ qubits
- The Toffoli gate is probably the best known of these gates
- Also known as the controlled-controlled-NOT gate, the Toffoli gate acts on three qubits: two control qubits and one target
- If both control qubits are set, then the amplitudes of the target qubit are flipped:



# Toffoli continued



- The Toffoli gate was originally devised as a universal, reversible *classical* logic gate by Tommaso Toffoli
- It is especially interesting because depending on the input, the gate can perform logical AND, XOR, NOT and FANOUT operations...
- This makes it universal for classical computing!
- Quantum computing is reversible:
  - All evolution in a quantum system can be described by unitary matrices, all unitary transformations are invertible, and thus all quantum computation is reversible
- The Toffoli gate implies that quantum computation is at least as powerful as classical computation

# Outline

# Classical computational complexity: a review

- To understand the possible power of quantum computing, it helps to look at the computational power of quantum computers in relation to their classical counterparts
- Remember that problems in P are decision problems that can be solved in polynomial time by a deterministic Turing machine
- The equivalent class for space efficiency is referred to as PSPACE
- NP problems are those that require a nondeterministic Turing machine in order to be solved efficiently
- The class of NP-complete problems, abbreviated NPC, consists of the hardest problems in NP
  - Every problem in NP can be reduced to a problem in NPC
  - If one NPC problem was found to be in P, then all of the problems in NP would also be in P, proving $P = NP$
  - Most theoretical computer scientests believe that $P \neq NP$, but nobody has been successful in proving the conjecture either way.

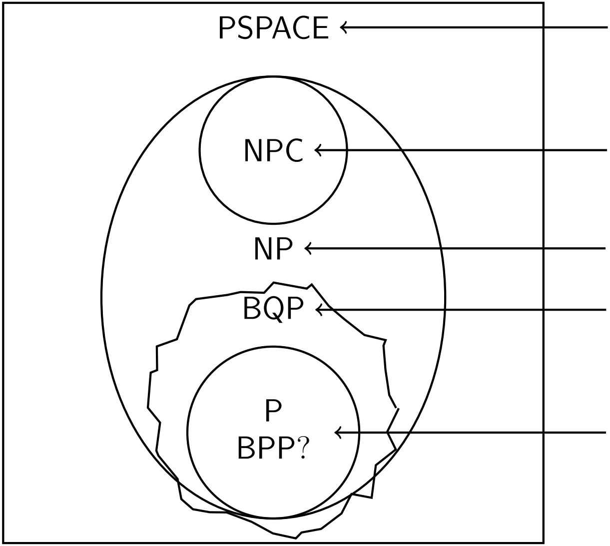
# Classical probabilistic complexity

- There is another important complexity class called BPP: Bounded-error Probabilistic Polynomial time
- BPP describes decision problems that can be solved in polynomial time by a *probabilistic* Turing machine
- Probabilistic Turing machines are those with direct access to some source of truly random input
- In BPP, the error of the solution is bounded in that the probability that the answer is correct must be at least two-thirds
- Although there are currently problems solvable in BPP that are not in P, the number of such problems has been decreasing since the introduction of BPP in the 1970's
- While it is not yet been proven whether $\mathrm{P} \subset \mathrm{BPP}$, it is conjectured that $\mathrm{P} = \mathrm{BPP}$

# Quantum computational complexity

- Quantum computation introduces a number of new complexity classes to the polynomial hierarchy
- Probably the most studied complexity class is Bounded-error Quantum Polynomial time, or BQP
- BQP is the quantum extension of BPP: the class of decision problems solvable in polynomial time by an innately probabilistic quantum Turing machine, with the same error constraint as defined for BPP
- Unlike BPP, it is suspected that $P \subset BQP$, which would mean that quantum computers are capable of solving some problems in polynomial time that cannot be solved efficiently by a classical Turing machine!

# A conjectured polynomial hierarchy




$n \times n$ chess
$n \times n$ Go

boolean satisfiability
subset sum
traveling salesman
map coloring
$n \times n$ sudoku

graph isomorphism

factoring
discrete logarithm

primality testing
graph connectivity
maximum matching
GCD

# Image Credits

- Quantum Computer: http://www.wired.com/wiredscience/2010/01/quantum-computer-hydrogen-simulation/
- Richard Feynman: http://www-scf.usc.edu/~kallos/feynman.htm
- Peter Shor: http://www-math.mit.edu/~shor/
- Cooling system for D-wave's quantum computer: http://mail2web.com/blog/wp-content/uploads/2007/03/d-wave-quantum-computer-cryopump.png
- Shrodinger's cat: http://confidentlysingle.com/2010/10/schrodingers-cat/
- David Deutsch: http://datapeak.net/computerscientists.htm
- Pauli & friends: http://scienceblogs.com/startswithabang/2010/10/the_story_of_the_neutrino.php
- Tommaso Toffoli: http://pm1.bu.edu/~tt/
- John T. Gill III: http://riddles.stanford.edu/gill/