Quantum Computers

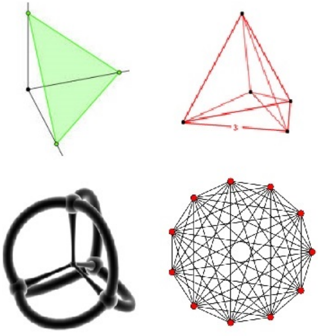
And How Does Nature Compute?

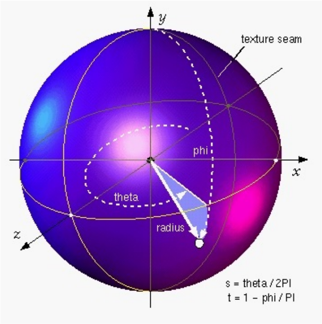
Kenneth W. Regan[1]
University at Buffalo (SUNY)

21 May, 2015

[1] Includes joint work with Amlan Chakrabarti, U. Calcutta

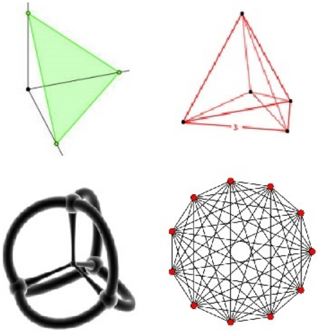
If you were designing Nature, how would you embody probabilities?

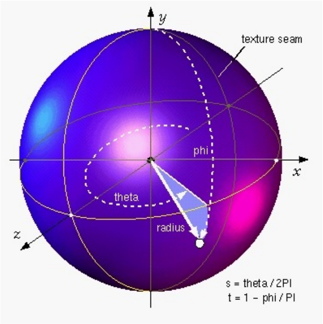




Simplex: $\sum_i p_i = 1$, each $p_i \geq 0$.
Spiky.

If you were designing Nature, how would you embody probabilities?

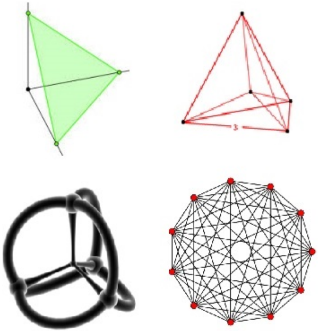


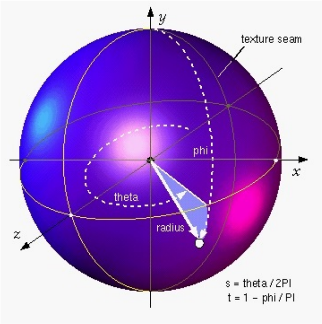


Simplex: $\sum_i p_i = 1$, each $p_i \geq 0$.
Spiky.

Sphere: $\sum_i |a_i|^2 = 1$; $p_i = |a_i|^2$.
Smooth.

If you were designing Nature, how would you embody probabilities?

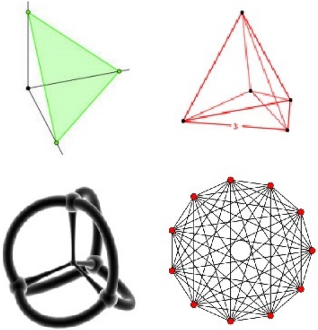


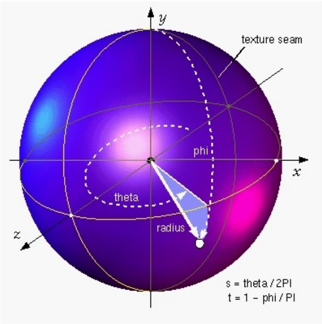


Simplex: $\sum_i p_i = 1$, each $p_i \geq 0$.
Spiky. Understood about 1950.

Sphere: $\sum_i |a_i|^2 = 1$; $p_i = |a_i|^2$.
Smooth.

If you were designing Nature, how would you embody probabilities?



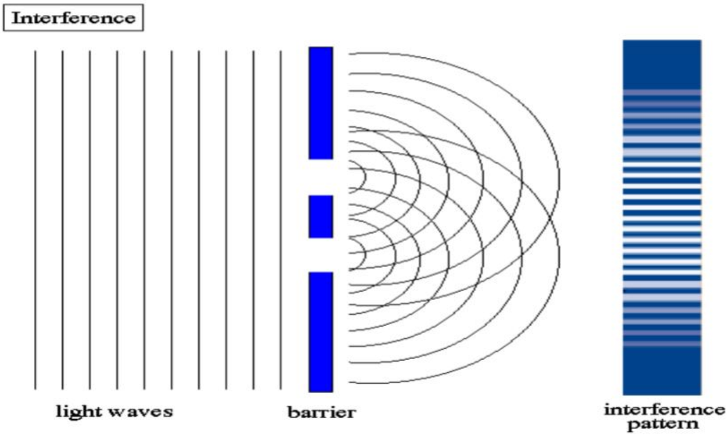


Simplex: $\sum_i p_i = 1$, each $p_i \geq 0$.
Spiky. Understood about 1950.

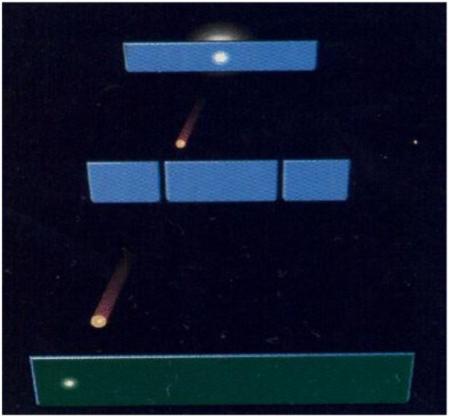
Sphere: $\sum_i |a_i|^2 = 1$; $p_i = |a_i|^2$.
Smooth. Understood by 300 BC.

Amplitudes and the Two-Slit Experiment

The a_i are called **amplitudes** and are physically real quantities.



…which works even when photons go singly!



Nature operates on the a_i. The probabilities p_i are "derivative." But why should Nature have probabilities at all?

Answer(?): She doesn't!

The Schrödinger Equation describes a deterministic process (simplified):

$$U(t) = e^{-iHt/\hbar}.$$

Here H is a time-independent operator on aggregates of amplitudes.

Answer(?): She doesn't!

The Schrödinger Equation describes a deterministic process (simplified):

$$U(t) = e^{-iHt/\hbar}.$$

Here H is a time-independent operator on aggregates of amplitudes. In the kind of discrete settings used for quantum computing, the aggregates are **state vectors** and the **Hamiltonian** operator H can be represented as an $N \times N$ matrix.

Answer(?): She doesn't!

The Schrödinger Equation describes a deterministic process (simplified):

$$U(t) = e^{-iHt/\hbar}.$$

Here H is a time-independent operator on aggregates of amplitudes. In the kind of discrete settings used for quantum computing, the aggregates are **state vectors** and the **Hamiltonian** operator H can be represented as an $N \times N$ matrix. The quantity e^{iA} is defined using the *eigenvalues* of the matrix A provided $A^2 = AA^* = I$ which makes them all real.

Answer(?): She doesn't!

The Schrödinger Equation describes a deterministic process (simplified):

$$U(t) = e^{-iHt/\hbar}.$$

Here H is a time-independent operator on aggregates of amplitudes. In the kind of discrete settings used for quantum computing, the aggregates are **state vectors** and the **Hamiltonian** operator H can be represented as an $N \times N$ matrix. The quantity e^{iA} is defined using the *eigenvalues* of the matrix A provided $A^2 = AA^* = I$ which makes them all real.

When H has cosmic scale this describes a multi-branch evolution, of which we experience one branch with statistical regularities that we experience as probabilities.

Answer(?): She doesn't!

The Schrödinger Equation describes a deterministic process (simplified):

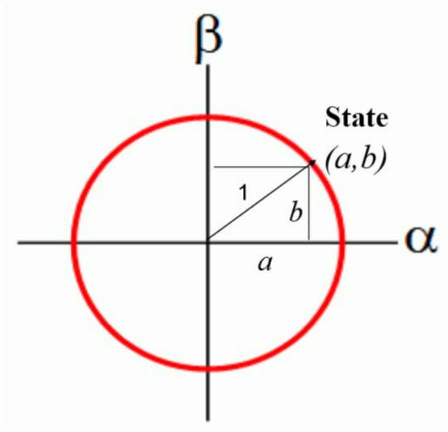
$$U(t) = e^{-iHt/\hbar}.$$

Here H is a time-independent operator on aggregates of amplitudes. In the kind of discrete settings used for quantum computing, the aggregates are **state vectors** and the **Hamiltonian** operator H can be represented as an $N \times N$ matrix. The quantity e^{iA} is defined using the *eigenvalues* of the matrix A provided $A^2 = AA^* = I$ which makes them all real.

When H has cosmic scale this describes a multi-branch evolution, of which we experience one branch with statistical regularities that we experience as probabilities. When H has tiny scale and $N = 2$ we get a **qubit**.

A Qubit

Quantum Bits, e.g. spins.



Probability of observing Alpha is a-squared, Beta is b-squared. By Pythagoras, these add to 1.

What if we have 17 qubits?

If the qubits are independent, you could represent their state by

$$(a_1, b_1), (a_2, b_2), (a_3, b_3), \ldots, (a_{17}, b_{17})$$

neatly using 34 entries.

What if we have 17 qubits?

If the qubits are independent, you could represent their state by

$$(a_1, b_1), (a_2, b_2), (a_3, b_3), \ldots, (a_{17}, b_{17})$$

neatly using 34 entries. To get the *amplitude* of any combination of states just multiply, e.g. for $x = 011 \cdots 0$:

$$e_x = e_{011\ldots0} \quad \text{has amplitude} \quad a_x = a_1 b_2 b_3 \cdots a_{17}.$$

What if we have 17 qubits?

If the qubits are independent, you could represent their state by

$$(a_1, b_1), (a_2, b_2), (a_3, b_3), \ldots, (a_{17}, b_{17})$$

neatly using 34 entries. To get the *amplitude* of any combination of states just multiply, e.g. for $x = 011 \cdots 0$:

$$e_x = e_{011\ldots0} \quad \text{has amplitude} \quad a_x = a_1 b_2 b_3 \cdots a_{17}.$$

You can "precompute" all $2^{17} = 131,072$ combinations by a vector of length $N = 131,072$ defined as the **tensor product** of the little vectors:

$$\vec{a} = (a_1, b_1) \otimes (a_2, b_2) \otimes (a_3, b_3) \otimes \cdots \otimes (a_{17}, b_{17}).$$

What if we have 17 qubits?

If the qubits are independent, you could represent their state by

$$(a_1, b_1), (a_2, b_2), (a_3, b_3), \ldots, (a_{17}, b_{17})$$

neatly using 34 entries. To get the *amplitude* of any combination of states just multiply, e.g. for $x = 011 \cdots 0$:

$$e_x = e_{011\ldots0} \quad \text{has amplitude} \quad a_x = a_1 b_2 b_3 \cdots a_{17}.$$

You can "precompute" all $2^{17} = 131,072$ combinations by a vector of length $N = 131,072$ defined as the **tensor product** of the little vectors:

$$\vec{a} = (a_1, b_1) \otimes (a_2, b_2) \otimes (a_3, b_3) \otimes \cdots \otimes (a_{17}, b_{17}).$$

Must *we* do this? Apparently *yes* if we wish to reckon with **entangled** states, which are definable as N-vectors that cannot be decomposed in this way.

What if we have 17 qubits?

If the qubits are independent, you could represent their state by

$$(a_1, b_1), (a_2, b_2), (a_3, b_3), \ldots, (a_{17}, b_{17})$$

neatly using 34 entries. To get the *amplitude* of any combination of states just multiply, e.g. for $x = 011 \cdots 0$:

$$e_x = e_{011\ldots0} \quad \text{has amplitude} \quad a_x = a_1 b_2 b_3 \cdots a_{17}.$$

You can "precompute" all $2^{17} = 131,072$ combinations by a vector of length $N = 131,072$ defined as the **tensor product** of the little vectors:

$$\vec{a} = (a_1, b_1) \otimes (a_2, b_2) \otimes (a_3, b_3) \otimes \cdots \otimes (a_{17}, b_{17}).$$

Must *we* do this? Apparently *yes* if we wish to reckon with **entangled** states, which are definable as N-vectors that cannot be decomposed in this way. *Does Nature do this?* That's the \$64,000,000,000 question. . .

Chalkboard Interlude...

[In the talk I illustrated nondeterministic and deterministic finite automata accepting the languages L_k of binary strings whose k-th from last bit is a 1. The NFA for L_3 needs only 4 states plus a dead state. The minimum DFA for L_3 needs $2^3 = 8$ states, and I drew all its twisted spreading on the board. For $k = 17$ the NFA grows only linearly to 18 states, but the DFA explodes to $2^{17} = 131,072$ states.

Again I posed the question: would we do the DFA or the NFA? What would Nature do? Well I could definitely say what UNIX does with `grep` and Perl and Python similaly when matching length-n lines of text to regular expressions: they build and simulate directly the **NFA**, taking $O(nk)$ time as opposed to $2^k n$ time.

I have not yet fully developed the NFA/DFA analogy to the "wave function of the universe"; reactions thus far are welcome.]

Allowed Operations

Nature allows any linear operation on state vectors that can be represented as a **unitary** matrix A of complex numbers:

$$AA^* = A^*A = I.$$

Then Ax always has the same length as x.

Allowed Operations

Nature allows any linear operation on state vectors that can be represented as a **unitary** matrix A of complex numbers:

$$AA^* = A^*A = I.$$

Then Ax always has the same length as x. For a tricky example, let

$$V = \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}.$$

Allowed Operations

Nature allows any linear operation on state vectors that can be represented as a **unitary** matrix A of complex numbers:

$$AA^* = A^*A = I.$$

Then Ax always has the same length as x. For a tricky example, let

$$V = \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}.$$

It is symmetric, so $V^T = V$, but complex conjugation makes a difference:

$$V^* = \frac{1}{2}\begin{bmatrix} 1-i & 1+i \\ 1+i & 1-i \end{bmatrix}.$$

Allowed Operations

Nature allows any linear operation on state vectors that can be represented as a **unitary** matrix A of complex numbers:

$$AA^* = A^*A = I.$$

Then Ax always has the same length as x. For a tricky example, let

$$V = \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}.$$

It is symmetric, so $V^T = V$, but complex conjugation makes a difference:

$$V^* = \frac{1}{2}\begin{bmatrix} 1-i & 1+i \\ 1+i & 1-i \end{bmatrix}.$$

Using $(1+i)(1-i) = 2$ but $(1+i)(1+i) = 2i$ which cancels $(1-i)(1-i) = -2i$, we get

$$V \cdot V^* = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{but} \quad V \cdot V = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

With Two Qubits

For $n = 2$ qubits you need $N = 2^n = 4$ as the vector and matrix dimension. Consider

$$U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix}$$

The column vector $e_{00} = (1, 0, 0, 0)^T$ stands for the "off-off" state, Then

$$Ue_{00} = \frac{1}{\sqrt{2}}(1, 0, 0, 1)^T = \frac{1}{\sqrt{2}}(e_{00} + e_{11}).$$

This means you have probability $1/2$ of *observing* 00 or 11 as outcomes, but will *never* observe 01 or 10. The two components are entangled.

More Qubits

The $\otimes$ product of vectors is a special case of the $\otimes$ product of matrices:

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots & a_{1,N}B \\ a_{2,1}B & a_{2,2}B & \cdots & a_{2,N}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1}B & a_{N,2}B & \cdots & a_{N,N}B \end{bmatrix}$$

More Qubits

The $\otimes$ product of vectors is a special case of the $\otimes$ product of matrices:

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots & a_{1,N}B \\ a_{2,1}B & a_{2,2}B & \cdots & a_{2,N}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1}B & a_{N,2}B & \cdots & a_{N,N}B \end{bmatrix}$$

If we do this n times with the 2×2 **Hadamard matrix**

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

then we get the **Hadamard transform** $H_N = H^{\otimes n}$.

More Qubits

The $\otimes$ product of vectors is a special case of the $\otimes$ product of matrices:

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots & a_{1,N}B \\ a_{2,1}B & a_{2,2}B & \cdots & a_{2,N}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1}B & a_{N,2}B & \cdots & a_{N,N}B \end{bmatrix}$$

If we do this n times with the 2×2 **Hadamard matrix**

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

then we get the **Hadamard transform** $H_N = H^{\otimes n}$. On argument $e_{00\ldots0}$ it produces the maximally superposed state

$$\frac{1}{\sqrt{2^n}}(1, 1, 1, \ldots, 1) = \frac{1}{\sqrt{2}}(1, 1) \otimes \cdots \otimes \frac{1}{\sqrt{2}}(1, 1).$$

Quantum Fourier Transform

With $\omega = e^{2\pi i/N}$, the ordinary Fourier matrix F_N is:

$$\frac{1}{\sqrt{N}}\begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{N-2} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{N-3} \\ \vdots & & & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{N-2} & \omega^{N-3} & \cdots & \omega \end{bmatrix}$$

That is, $F_N[i,j] = \omega^{ij \bmod N}$. As a "piece of code," it's simple.

What's "quantum" is the assertion that Nature provides sufficiently close approximations to this with about order-n^2 effort when $N = 2^n$. (Note also $F_N e_{00\cdots0} = H_N e_{00\cdots0}$.)

Shor's Algorithm

- It was known to Fourier that the Fourier transform converts *periodic* data into *concentrated* data.

Shor's Algorithm

- It was known to Fourier that the Fourier transform converts *periodic* data into *concentrated* data.
- A function of the form $f(x) = a^x \bmod M$ will be periodic with some period r, depending on both a and the prime factorization of M.

Shor's Algorithm

- It was known to Fourier that the Fourier transform converts *periodic* data into *concentrated* data.
- A function of the form $f(x) = a^x \bmod M$ will be periodic with some period r, depending on both a and the prime factorization of M.
- If we can learn r, then it was known pre-quantum that we can factor M.

Shor's Algorithm

- It was known to Fourier that the Fourier transform converts *periodic* data into *concentrated* data.
- A function of the form $f(x) = a^x \bmod M$ will be periodic with some period r, depending on both a and the prime factorization of M.
- If we can learn r, then it was known pre-quantum that we can factor M.
- M has exponential size in its number n of digits—and usually so does r—but using quantum we can probe that magnitude.

Shor's Algorithm

- It was known to Fourier that the Fourier transform converts *periodic* data into *concentrated* data.
- A function of the form $f(x) = a^x \bmod M$ will be periodic with some period r, depending on both a and the prime factorization of M.
- If we can learn r, then it was known pre-quantum that we can factor M.
- M has exponential size in its number n of digits—and usually so does r—but using quantum we can probe that magnitude.
- The "fuss" in Shor's algorithm is that we need to use a power of 2, $Q = 2^q$, with $Q \approx M^2$ and use binary approximation since r usually won't be a power of 2. But that's the idea.

Shor's Algorithm

- It was known to Fourier that the Fourier transform converts *periodic* data into *concentrated* data.
- A function of the form $f(x) = a^x \bmod M$ will be periodic with some period r, depending on both a and the prime factorization of M.
- If we can learn r, then it was known pre-quantum that we can factor M.
- M has exponential size in its number n of digits—and usually so does r—but using quantum we can probe that magnitude.
- The "fuss" in Shor's algorithm is that we need to use a power of 2, $Q = 2^q$, with $Q \approx M^2$ and use binary approximation since r usually won't be a power of 2. But that's the idea.
- Factoring numbers M allows breaking the RSA cryptosystem with effort roughly $O(n^3)$, whereas the best known on classical computers is roughly $2^{n^{1/3}}$.

Efforts to Build Quantum Computers

[At this point I showed webpages to discuss the current state, 21 years on from Shor's algorithm. Outline:

Efforts to Build Quantum Computers

[At this point I showed webpages to discuss the current state, 21 years on from Shor's algorithm. Outline:

- Just before the millennium, Shor's algorithm was demonstrated by factoring $15 = 5 \times 3$ using 4 main qubits and a few "ancilla" (helper) qubits.

Efforts to Build Quantum Computers

[At this point I showed webpages to discuss the current state, 21 years on from Shor's algorithm. Outline:

- Just before the millennium, Shor's algorithm was demonstrated by factoring $15 = 5 \times 3$ using 4 main qubits and a few "ancilla" (helper) qubits.
- Not much progress has been made since... and perhaps even those demos "cheated" a bit (http://arxiv.org/abs/1301.7007).

Efforts to Build Quantum Computers

[At this point I showed webpages to discuss the current state, 21 years on from Shor's algorithm. Outline:

- Just before the millennium, Shor's algorithm was demonstrated by factoring $15 = 5 \times 3$ using 4 main qubits and a few "ancilla" (helper) qubits.
- Not much progress has been made since…and perhaps even those demos "cheated" a bit (http://arxiv.org/abs/1301.7007).
- Why are we finding it so hard to "scale up" quantum computers?

Efforts to Build Quantum Computers

[At this point I showed webpages to discuss the current state, 21 years on from Shor's algorithm. Outline:

- Just before the millennium, Shor's algorithm was demonstrated by factoring $15 = 5 \times 3$ using 4 main qubits and a few "ancilla" (helper) qubits.
- Not much progress has been made since... and perhaps even those demos "cheated" a bit (http://arxiv.org/abs/1301.7007).
- Why are we finding it so hard to "scale up" quantum computers?
- I moderated a debate on the "Gödel's Lost Letter" blog between Gil Kalai and Aram Harrow, all during 2012. Richard Lipton and I are beginning to update it for a book.

Efforts to Build Quantum Computers

[At this point I showed webpages to discuss the current state, 21 years on from Shor's algorithm. Outline:

- Just before the millennium, Shor's algorithm was demonstrated by factoring $15 = 5 \times 3$ using 4 main qubits and a few "ancilla" (helper) qubits.
- Not much progress has been made since... and perhaps even those demos "cheated" a bit (http://arxiv.org/abs/1301.7007).
- Why are we finding it so hard to "scale up" quantum computers?
- I moderated a debate on the "Gödel's Lost Letter" blog between Gil Kalai and Aram Harrow, all during 2012. Richard Lipton and I are beginning to update it for a book.
- At the heart are schemes for **quantum error-correcting codes**, also partly originated by Shor, and the Quantum Fault Tolerance Theorem giving an absolute physical threshold which if met by the raw **decoherence** error rate enables the codes to succeed.

Current Quantum Devices and Efforts / Conclusion

Current Quantum Devices and Efforts / Conclusion

- The **D-Wave** company can harness 100–500 qubits in a short-lived *adiabatic* process claimed to yield output speeding up numerical computations that is hard to achieve "classically." Debate about these claims, still short of being "quantum universal," is ongoing.

Current Quantum Devices and Efforts / Conclusion

- The **D-Wave** company can harness 100–500 qubits in a short-lived *adiabatic* process claimed to yield output speeding up numerical computations that is hard to achieve "classically." Debate about these claims, still short of being "quantum universal," is ongoing.
- The **Boson Sampling** idea is promising but also sub-universal(!?).

Current Quantum Devices and Efforts / Conclusion

- The **D-Wave** company can harness 100–500 qubits in a short-lived *adiabatic* process claimed to yield output speeding up numerical computations that is hard to achieve "classically." Debate about these claims, still short of being "quantum universal," is ongoing.
- The **Boson Sampling** idea is promising but also sub-universal(!?).
- Efforts at truly universal quantum computers are being ramped up all over the world, most publicly in Europe.

Current Quantum Devices and Efforts / Conclusion

- The **D-Wave** company can harness 100–500 qubits in a short-lived *adiabatic* process claimed to yield output speeding up numerical computations that is hard to achieve "classically." Debate about these claims, still short of being "quantum universal," is ongoing.
- The **Boson Sampling** idea is promising but also sub-universal(!?).
- Efforts at truly universal quantum computers are being ramped up all over the world, most publicly in Europe.
- Will they work? Lipton and I voice skepticism; moreover we believe factoring can be done efficiently without a quantum computer.

Current Quantum Devices and Efforts / Conclusion

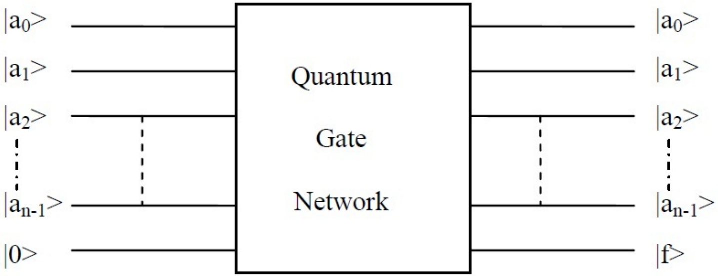
- The **D-Wave** company can harness 100–500 qubits in a short-lived *adiabatic* process claimed to yield output speeding up numerical computations that is hard to achieve "classically." Debate about these claims, still short of being "quantum universal," is ongoing.
- The **Boson Sampling** idea is promising but also sub-universal(!?).
- Efforts at truly universal quantum computers are being ramped up all over the world, most publicly in Europe.
- Will they work? Lipton and I voice skepticism; moreover we believe factoring can be done efficiently without a quantum computer.
- To conclude: factoring and breaking RSA follow **if we can find *human* notation for how *Nature* "really" computes.**

Current Quantum Devices and Efforts / Conclusion

- The **D-Wave** company can harness 100–500 qubits in a short-lived *adiabatic* process claimed to yield output speeding up numerical computations that is hard to achieve "classically." Debate about these claims, still short of being "quantum universal," is ongoing.
- The **Boson Sampling** idea is promising but also sub-universal(!?).
- Efforts at truly universal quantum computers are being ramped up all over the world, most publicly in Europe.
- Will they work? Lipton and I voice skepticism; moreover we believe factoring can be done efficiently without a quantum computer.
- To conclude: factoring and breaking RSA follow **if we can find *human* notation for how *Nature* "really" computes.**
- My own research tries to find Nature's secret in the algebra of multi-variable polynomials, into which **quantum circuits** can be translated. A more-technical version of the talk would include the following slides on quantum circuits, then show my blog article rjlipton.wordpress.com/2012/07/08/grilling-quantum-circuits/ .]

Quantum Circuits

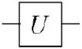
Quantum circuits look more constrained than Boolean circuits:



But Boolean circuits look similar if we do Savage's TM-to-circuit simulation and call each *column* for each tape cell a "cue-bit."

Quantum gates

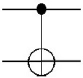
single qubit operation:



controlled-NOT:

control

target

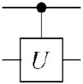


unitary matrix $= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

controlled-U:

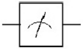
control

target



unitary matrix $= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{00} & U_{01} \\ 0 & 0 & U_{10} & U_{11} \end{pmatrix}$

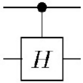
measurement in the $|0\rangle, |1\rangle$ **basis:**



[Slides concept by D. Bacon, U Washington]

Quantum gates: an example

controlled-gate
(here controlled-H)

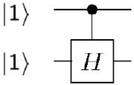


$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

input: $|\psi\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle$

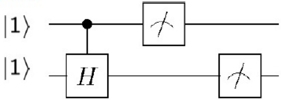
output: $|\psi'\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$

compute:



$$= \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} = \frac{1}{\sqrt{2}}(|10\rangle - |11\rangle)$$

measure:



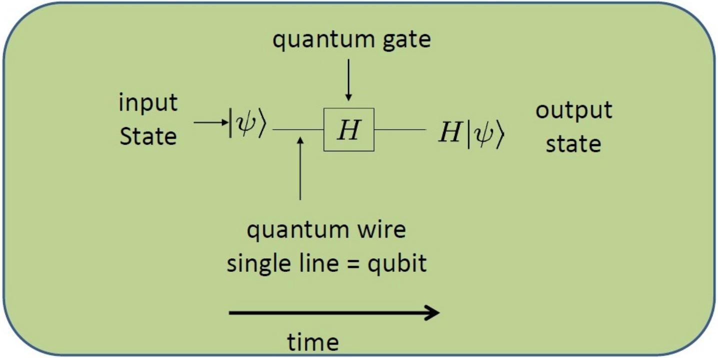
Probability of 10: $\left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}$

Probability of 11: $\left|\frac{-1}{\sqrt{2}}\right|^2 = \frac{1}{2}$

Probability of 00 and 01: $|0|^2 = 0$

Quantum circuits

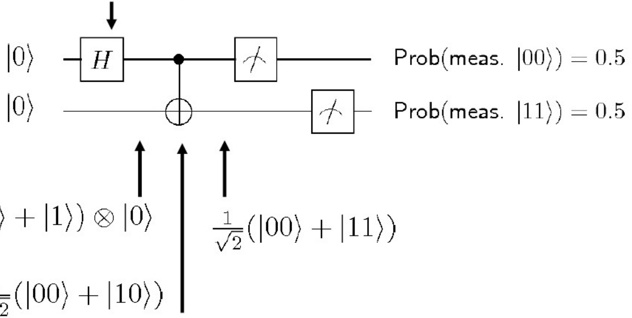
Quantum circuit diagrams to visualize a computation:



Quantum circuits are sequences of instructions. Describes a series of unitary evolutions (quantum gates) applied to a quantum state.

Quantum circuit example

$$H \otimes \mathbf{1}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \mathbf{1}_2$$



$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$$

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

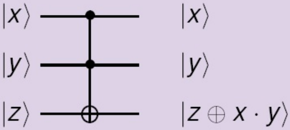
$$\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$$

$$|00\rangle \rightarrow |00\rangle$$
$$|01\rangle \rightarrow |01\rangle$$
$$|10\rangle \rightarrow |11\rangle$$
$$|11\rangle \rightarrow |10\rangle$$

Toffoli Gate

The Toffoli gate "TOF"

x	y	z	x'	y'	z'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0



Theorem (Toffoli, 1981)

Any reversible computation can be realized by using TOF gates and ancilla (auxiliary) bits which are initialized to 0.

Slides by Martin Rötteler

Bounded-error Quantum Poly-Time

A language A belongs to BQP if there are uniform poly-size quantum circuits C_n with n data qubits, plus some number $\alpha \geq 1$ of "ancilla qubits," such that for all n and $x \in \{0, 1\}^n$,

$$\begin{aligned} x \in A &\implies \Pr[C_n \text{ given } \langle x0^\alpha| \text{ measures 1 on line } n+1] > 2/3; \\ x \notin A &\implies \Pr[\dots] < 1/3. \end{aligned}$$

One can pretend $\alpha = 0$ and/or measure line 1 instead. One can also represent the output as the "triple product" $\langle a \mid C \mid b \rangle$, with $a = x0^\alpha$, $b = 0^{n+\alpha}$.

Two major theorems about BQP are:

(a) C_n can be composed of just Hadamard and Toffoli gates [Y. Shi].

(b) Factoring is in BQP [P. Shor].

[Segue to "Grilling Quantum Circuits" post on GLL blog.]