Reversible Computation

Second law of thermodynamics: Entropy never decreases

Landauer's principle, "any logically irreversible manipulation of information, such as the erasure of a bit or the merging of two computation paths, must be accompanied by a corresponding entropy increase in non-information bearing degrees of freedom of the information processing apparatus or its environment". The minimum possible amount of energy required to change one bit of information, known as the Landauer limit: kT ln 2, where k is the Boltzmann constant (approximately 1.38×10−23 J/K), T is the temperature of the circuit in kelvins, and ln 2 is the natural logarithm of 2 (approximately 0.69315).

But all computation can be made reversible (time invertible):

For a Turing Machine: At a cost of squaring the time of the computation, one can just write down a computation history as one is computing

For circuits: The basic Fredkin gate is a controlled swap gate that maps three inputs (*C_in*, I_1, I_2) onto three outputs (*C_out*, O_*1*, *O*_2). The *C* input is mapped directly to the *C* output. If *C* = 0, no swap is performed; *I*_1 maps to *O*_1, and *I*_2 maps to *O*_2. Otherwise, the two outputs are swapped so that *I*_1 maps to *O*_2, and *I*_2 maps to *O*_1. It is easy to see that this circuit is reversible, i.e., "undoes" itself when run backwards. It has the useful property that the numbers of 0s and 1s are conserved throughout, which in the billiard ball model means the same number of balls are output as input. This corresponds nicely to the conservation of mass in physics, and helps to show that the model is not wasteful.

O_1= (NOT C_in AND I_1) OR (C_in AND I_2)
O_2 = (C_in AND I_1) OR (NOT C_in AND I_2)
C_out= C_in

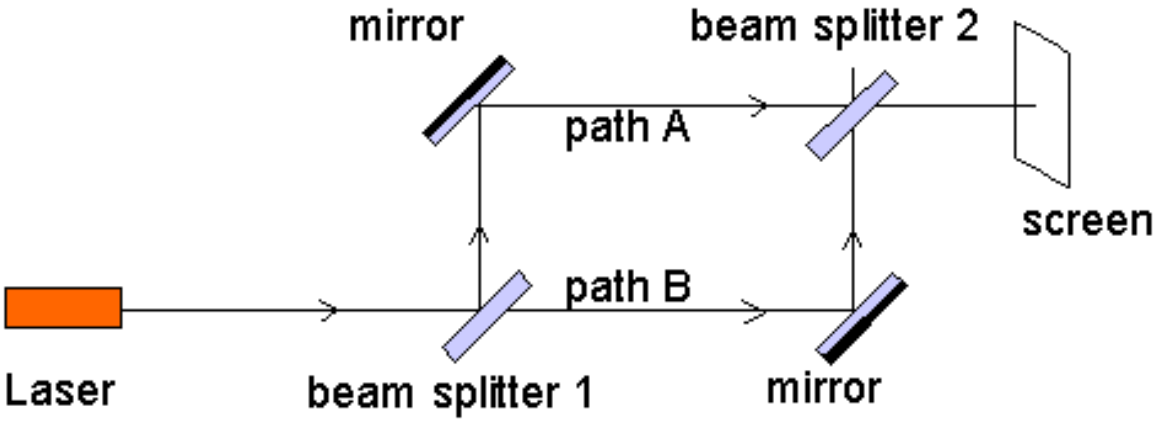
One way to see that the Fredkin gate is universal is to observe that it can be used to implement AND,NOT and OR:

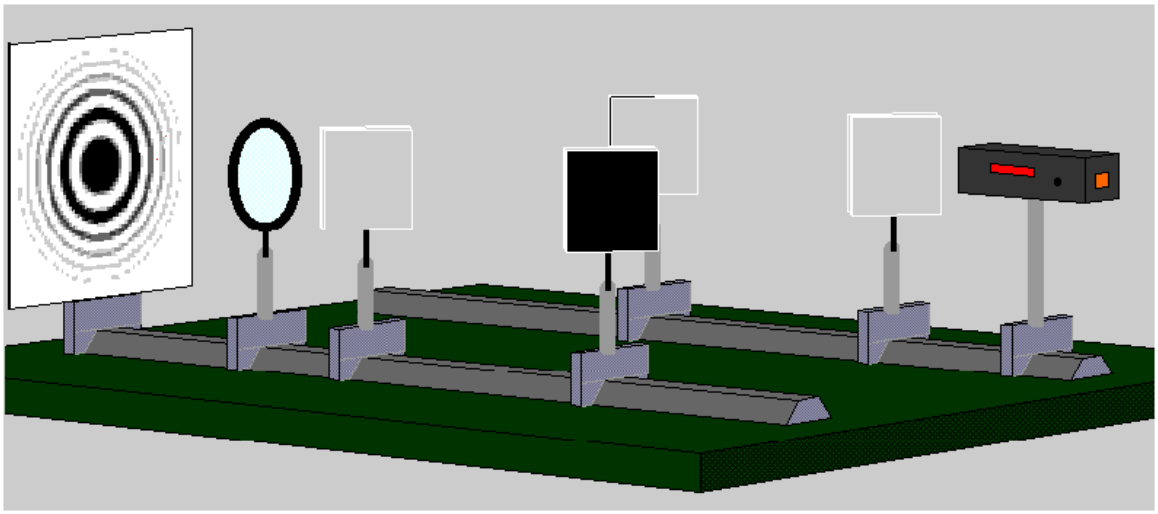
If I_1 = 0 then O_1 = C AND I_2

If I_2= 0 and I_1 = 1 then O_1 = NOT C_in

Quantum Notes

Half-Silver mirror experiment





First do one beam splitter (half silvered mirror) to show that randomness exists. Then do the full experiment. Then discuss what happens if you block a path

Show computation tree with four leaves, and discuss probabilities and computing.

Definition states: If a system can be in k states, then you can think of the kth state as the unit column vector with a 1 in the ith row, and zeros every place else. More generally, the possible states of a system form an orthnormal basis. So in the above experiment, there are two states

|H> = Horizonal moving = |1| and |V> = vertical moving = |0|

|0| |1|

Superposition principle: system may be in a superimposed state

a |H> + b |V>

where a^2 + b^2 = 1 (i.e. norm of the vector is unit). The coefficients a and b may be complex.

Definition of Unitary operation (e.g. the mirrors): Linear invertible/reversible operation that maps unitary vectors to unitary vectors. Operation can then be expressed as k by k matrix. Note that quantum operations are necessarily reversible. So this ties quantum operations necessarily have no minimum energy requirement to perform the computation.

Operation matrix for ½ silver mirror

| 1/sqrt(2) 1/sqrt(2) |

| 1/sqrt(2) -1/sqrt(2)|

Consider applying the operation to a H photon

| 1/sqrt(2) 1/sqrt(2) | |1| = | 1/sqrt(2) |

| 1/sqrt(2) -1/sqrt(2)| |0| |1/sqrt(2) |

Operation matrix for full silver mirror (not gate)

| 0 1|

| 1 0|

Counter intuitively, the not operation doesn't change the state of the particle.

Now consider what happens applying the second silver mirror

| 1/sqrt(2) 1/sqrt(2) | || 1/sqrt(2) | = |1| = H

| 1/sqrt(2) -1/sqrt(2)| |1/sqrt(2) | |0|

Definition of Measurement: Recall that what you are measuring is a state, and that the states are an orthonormal basis. The probability that you observe basis vector b when in supposition state s is the inner product of s and b squared

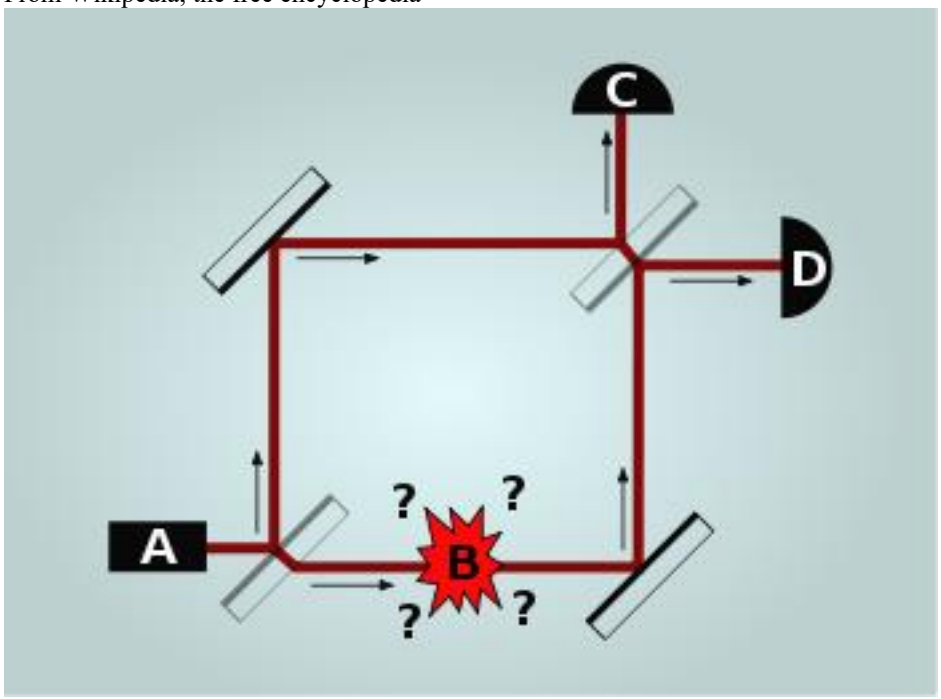
Example: The photon after 1 half silver gate

Example: The photon after 2 half silver gates

Note that a measurement and an operator are two different sorts of beasts. Quantum mechanics is mum on what causes a measurement, although any macroscopic action that intuitively constitutes a measurement causes a measurement in the quantum mechanical sense.

Elitzur–Vaidman bomb tester

From Wikipedia, the free encyclopedia



Bomb-testing problem diagram. A - photon emitter, B - bomb to be tested, C,D - photon detectors. Mirrors in the lower left and upper right corners are half-silvered.

In physics, the **Elitzur–Vaidman bomb-testing problem** is a thought experiment in quantum mechanics, first proposed by Avshalom Elitzur and Lev Vaidman in 1993.[1] An actual experiment demonstrating the solution was constructed and successfully tested by Anton Zeilinger, Paul Kwiat, Harald Weinfurter, and Thomas Herzog from the University of Innsbruck, Austria and Mark A. Kasevich of Stanford University in 1994.[2] It employs a Mach–Zehnder interferometer to check if a measurement has taken place.

Problem

Consider a collection of bombs, of which some are duds. Suppose each usable (non-dud) bomb has a photon-triggered sensor, which will absorb an incident photon and detonate the bomb. Dud bombs have no sensor, so do not interact with the photons. Thus, the dud bomb will not detect the photon and will not detonate. Is it possible to detect if a bomb is a non-dud without detonating it? Is it possible to determine that some bombs are non-duds without detonating all of them?

Solution

A bomb is placed on the lower path of a Mach–Zehnder interferometer with a single-photon light source. If the photon takes the lower path and the bomb is live, then the photon is absorbed and triggers the bomb; otherwise, if the bomb is a dud, the photon will pass through unaffected.

When a photon passes through a half-silvered plane mirror, it enters a quantum superposition of all possible outcomes, which interact with each other. The photon is both transmitted and reflected, and takes both paths through the interferometer. The interference from the two routes determines the probability of

detection at each detector (C and D). The photon remains in the superposition state until an observer (the bomb's photon sensor, if present, and later the detector at C or D) causes the wave function to collapse and the photon assumes a single one of the states.

The interferometer is aligned so that the interference is constructive at C and destructive at D. If the bomb is a dud, it does not affect the split wave, and photons will only ever be detected at C. If a live bomb is placed in the lower path, it blocks this route and so destroys the interference pattern, and the photon will have a 50% chance of being detected in either detector (but never both). Note that even if the live bomb does not actually detect the photon, it still performs a measurement of whether the photon travels along that path (a negative-result measurement, in this case), and therefore still guarantees that the photon only travels along the upper path.

Thus if a photon is detected in D there must be a live, photon-blocking bomb. If a photon is detected at C then the bomb may be either live or a dud. No photon is detected in the case of detonation (since the photon gets absorbed by the sensor), but the detonation will rattle the apparatus.

Once a detection has been made, the superposition is destroyed and the photon path becomes certain. If there is a live bomb, there is a 50% chance the photon takes the lower path and the bomb detonates. There is a 25% chance the photon takes the upper path at both mirrors and is detected at C, and a 25% chance the photon takes the upper path and is detected at D.

With this process 25% of live bombs can be identified without being detonated,[1] 50% will be detonated and 25% remain uncertain. By repeating the process with the uncertain ones, the ratio of identified, non-detonated live bombs approaches 33% of the initial population of bombs. See the "Experiments" section below for a modified experiment that can identify the live bombs with a yield rate approaching 100%.

Now consider removing the lower-right full silvered mirror. Now the one full-silvered mirror would seem to implement the operation |H> goes to |H> and |V> goes to |H>.

Question: Why isn't this possible?
Answer: Its not reversible. But this just means that the modeling with only |H> and |V> is too restrictive.

Tensor Product

If qubit |i> is in state ‘

a |0> + b |1>

and qubit |j> is independently in state

c |0> + d |1>

Then the pair of qubits is in state

$|ij> = |i> \otimes |j> = (a |0> + b |1>) \otimes (c |0> + d |1>) =$

ac |00> + ad |01> + bc |10> + bd |11>

Here $\otimes$ is called the tensor product, or maybe outer product.

Conditional Amplitudes

If

|ij> = a |00> + b |01> + c |10> + d |11>

And one observes the qubit i, then with probability $a^2 + c^2$ one sees a 0, and the resulting state is:

$|ij> = a/(a^2 + c^2)^{1/2} |00> + c/(a^2+c^2)^{1/2} |10>$

and with probability $b^2 + d^2$ one sees a 1, and the resulting state is:

$|ij> = b/(b^2 + d^2)^{1/2} |10> + d/(b^2+d^2)^{1/2} |11>$

Entangled Bits

Consider the gate G that has inputs i and j and outputs i and i XOR j.

Question: Is this gate reversible? Answer: yes

Assume i = |0>/sqrt(2) + |1>/sqrt(2) and j=0

Question: What is the output state?

Answer: |00>/sqrt(2) + |11>/sqrt(2)

Note that the states of these two bits are “entangled”, that is, a measurement one bit necessarily constitutes a measurement on the other bit.

EPR “Paradox”:

Parity Game: Alice and Bob are far apart. Although they may chat/communicate before they are separated. Alice is given a random bit x, and Bob is given a random bit y. Alice has to quickly produce a bit a, and Bob has to quickly produce a bit b. Since there is an upper bound on the speed of light, they can communicate after seeing x and y, and before producing a and b. Alice and Bob win if

x AND y = a XOR b

Question: What are some obvious strategies and their probability for winning?

One: Flip a coin for a and a coin for b. This give probability ½ of winning

Two: always set a=0 and b=0. It is clear that they win with probability ¾ with this strategy. A bit of reflection will reveal that they have no better strategy (without using quantum mechanics).

How to win with probability greater than ¾ using quantum mechanics.

Consider the following strategy:

Each of Alice and Bob takes one of these two entangled bits before they separate.

Alice: a = measure the entangled bit

Consider the truth table for the optimal Newtonian strategy:

x	y	x and y	a xor b
0	0	0	0
1	0	0	0
0	1	0	0
1	1	1	0

Question: From Alice’s (Bob’s) point of view, what value of x (y) is dangerous for this strategy?

Answer: 1. Then there is a 50/50 chance from that person's point of view that the answer will be wrong. So when x=1, Alice would like to like a chance to see something other than what Bob sees; So before observing, Alice rotates her bit Theta in one direction. Similarly, if Bob sees y=1, he rotates his bit Theta in the other direction.

Question: Which case does this hurt?
Answer: If only one of x and y is one.

Question: Why is hurt less than help?
Answer: If both x and y are 1, there is a 2Theta difference, but when only 1 of x and y is 1, there is only a Theta difference/error introduced.

Alice's Protocol: If x = 1 then rotate the state of her qubit by pi/8 and then measure the qubit. If x=0 then measure the qubit directly. The value of a is the result of the measurement.

A rotation operation by an angle theta can be represented by the matrix

|cos theta -sin theta |

|sin theta cos theta |

Assume that |0> =|1| and |1> = |0|

|0| |1|

Bob's Protocol: If y =1 then rotate the state of the qubit by –pi/8 and then measure the qubit. If y=0 then measure the qubit directly. The value of a is the result of the measurement.

Question: If x=y=0 then what is the probability that a=b?

Answer: 1

Question: If x <> y then what is the probability that a=b?

Answer: By symmetry, we can assume without loss of generality that Alice does the rotation. Before Alice does the rotation, the state is cos pi/4 |00> + sin pi/4 |11>.

By linearity, we can apply the rotation to each of the two states. Applying the rotation to the state |00> gives cos pi/8 |00> + sin pi/8 |10>. Applying the rotation to the state |11> gives cos 5pi/8 |01> + sin 5pi/8 |11>. Thus after the rotation, Alices bit is in state

Cos pi/4 cos pi/8 |00> + cos pi/4 sin pi/8 |10> - sin pi/4 sin pi/8 |01> + sin pi/4 cos pi/8 |11>

Let us say that Alice measures her bit before Bob (it doesn't matter). The probability that Alice measures a 0 is (Cos pi/4 cos pi/8)^2+(- sin pi/4 sin pi/8)^2 = .14

If Alice measures a 0, then the state of the system is cos pi/8 |00> - sin pi/8 |01>. Think that this is the probability conditioned on the fact that Alice's bit is 0. You need to scales so that the squared sum of the coefficients is 1. Now the probability that Bob measures his bit to be 0 is cos^2 pi/8 = .84

The probability that Alices measures a 1 is (cos pi/4 sin pi/8)^2 + (sin pi/4 cos pi/8)^2 = .85, If Alice measures a 1 then the state of the system is + sin pi/8 |10> + cos pi/8 |11>. Now the probability that Bob measures his bit to be 1 is cos^2 pi/8.

Thus independent of what Alice observes, Bob chances of observing the same bit value is cos^2 pi/8

Question: If x=y=1 what is the probability that a=b?

Answer: ½ using the same line of reasoning as above, but it is a bit more complicated because there are two rotations. I'll leave this as homework.

Quantum Computation model: On input I of size n and space m and time t, you start in state |I 0^(m-n)> and apply a sequence of t quantum operators, where each operator can only be applied to a constant number of bits. Measure the state at the end for your output. You need to get the answer you want with probability bounded away from ½.

Many quantum computations, including Simon's algorithm nd Shor's algorithm, have the following form:

1. Apply Hadamard operation some of the bits
2. Perform a sequence of reversible classical operations
3. Apply Hadamard operation to some of the bits
4. Measure/observe

The half silvered mirror is a 1 bit Hadamard gate/operation H_1. That is,

H_1 = | 1/sqrt(2) 1/sqrt(2) |

| 1/sqrt(2) -1/sqrt(2)|

The following are equivalent definitions of H_n:

1. H_1 applied independently to n bits
2. H_n = 1/sqrt(2) | H_{n-1} H_{n-1} |

   | H_{n-1} - H_{n-1} |
3. The (i,j) entry of H_n is 2^(-n/2) (-1)^(i*j) where (i*j) is the inner product of the bit vectors modulo 2

Simon's Algorithm. This is a polynomial time quantum algorithm for a problem that presumably takes exponential time for a deterministic/randomized algorithm. Let f be some function from 2^n to 2^n such that there exists an n bit string a such that f(x)=f(y) iff x = y xor a. For f is a 2 to 1 mapping. You are given f (say either as a black box, or as a circuit), you goal is to find a.

Deterministically/Randomly: You don't learn much until you find an x and y such that f(x)=f(y) [this is not completely trivial], so you need about 2^(n/2) queries (this is the birthday problem).

Quantum algorithm:

Let k=n/2.

Start in state |0^2n>

Appply a Hadamard gate to the first n bits to get to state

½^k sum_{x in 2^n} |x 0^n>

Apply the reversible operation |xz> goes to |x (z xor f(x)) >

½^k sum_{x in 2^n} |x f(x)> = ½^k sum_{x in 2^(n-1)} (|x> + |x xor a>) |f(x)>

Measure the second n bits of the memory to get state

(|x> + |x xor a>) |f(x)>

comment: So the second n bits are now fix, and the first n bits are entangled

and in a superposition between two options

Question: If we knew x and (x xor a) then we could deduce a. So why doesn't measuring the first n bits give us what we want?

Answer: If we measure the first n bits we would get x with probability ½ and x xor a with probability ½. In either case, we would lose the value of the other one in collapse. So we need to do something more subtle.

So we apply the Hadamard gate to the first n bits again to get

[½^k sum_{y in 2^n} ((-1)^(x*y) + (-1)^((x xor a)* y)) |y>] |f(x)> =

[½^k sum_{y in 2^n} ((-1)^(x*y) + (-1)^(x*y) (-1)^(a*y)) |y>] |f(x)> =

$2/2^k \sum_{y \in 2^n \text{ such that } a \cdot y = 0} |y\ f(x)\rangle$

So measure the first n bits we get a y uniformly at random such that $y \cdot a = 0$

Comment: Here the operation "*" is inner-product modulo 2.

Using some basic linear algebra, if we repeat this a linear number of times, we get enough information (linear equations) to determine a with high probability.

Lemma: [Quantum Teleportation] Given a pair of entangled bits, one can communicate a qubit by sending two classical bits

Lemma: [Superdense Coding] Given a pair of entangled bits, one can communicate two classical bits by sending one qubit

Resulting Theorem: If entangled bits are free, then information in 1 qubit = information in 2 classical bits

Quantum Teleportation

Same setup as EPR experiment.

- Alice and Bob split up entangled qubits |ij> = (|00> + |11)/sqrt(2).
- After being split up, Alice gets qubit |x> = a |0> + b |1>)
- Alice sends Bob 2 classical bits a, and b
- From a and b, Bob changes the state of |j> to a (|0> + b |1>)

Algorithm:

Initial state of |xij> = [a (|0> + b |1>)]⊗ [(|00> + |11)/sqrt(2)]

= (a |000> + b |100> + a |011> + b|111>) /sqrt(2)

Easiest to understand by first focusing in on the state of |xij> right before Alice sends classic bits a and b

$$|xij> = |00> \otimes \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} [a\,|0> + b\,|1>]/2 \right) +$$

$$|01> \otimes \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} [a\,|0> + b\,|1>]/2) \right) +$$

$$|10> \otimes \left(\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} [a\,|0> + b\,|1>]/2 \right) +$$

$$|11> \otimes \left(\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} [a\,|0> + b\,|1>]/2 \right)$$

From this state:

- Alice observes bits |xi>.
  - If Alice observes |00> then |j> = $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ [a |0> + b |1>] = a |0> + b |1>
  - If Alice observes |01> then |j> = $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$ [a |0> + b |1>)
  - If Alice observes |10> then |j> = $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ [a |0> + b |1>]
  - If Alice observes |11> then |j> = $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ [a |0> + b |1>]
- Alice sends now classical bits x and i to Bob
- Bob then
  - If xi = 00 then Bob sets |j> = (inverse $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$) |j> = a |0> + b |1>)
  - If xi = 01 then Bob sets |j> = (inverse $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$) |j> = a |0> + b |1>)
  - If xi = 10 then Bob sets |j> = (inverse $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$) |j> = a |0> + b |1>)
  - If xi = 11 then Bob sets |j> = (inverse $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$) |j> = a |0> + b |1>)

Note that Bob has then recovered |j> in the initial state of |x>. Note that Alice's copy of |x> has been destroyed, so this is transportation/teleportation, and not copying.

Now we are left to determine how Alice modifies |xi> to reach this desired intermediate state.

- Alice applies the reversible classical "controlled not" operation to x and i. So if x = 1 then i= not i else i=i
- Alice then applies the 1 bit Hadamard operation H_1 to |x>

After the controlled not |xij> is in state

(a |000> + b |110> + a |011> + b|101>) /sqrt(2)

After applying H_1 to |x> then |xij> is in state:

[a (|000> + |100>) +b (|010> - |110>) + a (|011> + |111>) + b (|001> - |101>)]/2

which one can verify by calculations is equal to the desired intermediate state, namely

$$|xij\rangle = |00\rangle \otimes \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} [a|0\rangle + b|1\rangle]/2 \right) +$$

$$|01\rangle \otimes \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} [a|0\rangle + b|1\rangle]/2 \right) +$$

$$|10\rangle \otimes \left(\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} [a|0\rangle + b|1\rangle]/2 \right) +$$

$$|11\rangle \otimes \left(\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} [a|0\rangle + b|1\rangle]/2) \right)$$

No Cloning Theorem: There is no quantum operation A that can map

$(a|0\rangle + b|1\rangle) \otimes |0\rangle$ to

$(a|0\rangle + b|1\rangle) \otimes (a|0\rangle + b|1\rangle)$

Proof:

Assume to reach a contradiction that such an A exists

By linearity $A\,(a|0\rangle + b|1\rangle) \otimes |0\rangle$

$= A\,(a|00\rangle + b|10\rangle)$

$= a|00\rangle + b|11\rangle$ by the property of A

But note this does not equal $(a|0\rangle + b|1\rangle) \otimes (a|0\rangle + b|1\rangle)$

Quantum Cryptography: Let XY = { |x> = [1 0], |y> = [0 1]} and AB = { |a> = [cos(pi/4), sin pi/4], |b> = [cos –pi/4, sin –pi/4] } be two properties.

Quantum Indeterminacy Principle: If you know the value of one of the properties AB or XY with certainty, then the other property must be in superposition.

Here is a public key cryptographic protocol that detects eaves dropping:

Alice: Send 4n particles uniformly at random in state x, y, a, b

Bob: For each particle flip a fair coin to determine whether to measure property AB or XY. Bob then tells Alice what property he measured for each particle.

Alice: Tells Bob what property she sent each particle in.

They both throw away particles where the sent and measured properties were not equal. This leaves them with out 2n bits.

Alice: Uniformly at random pick n particles names, and tells Bob what state you sent these particles in

Bob: Determines if his measured state matches Alice's sent state on all particles.

If there is a match on all n tests particles, then Alice and Bob use the remaining sqrt(n) bits as a shared secret key and then use private key cryptography. If there is a mismatch on any tested property, then Alice and Bob declare that there is an eavesdropper.

Proof that protocol is secure with high probability: If you are an eavesdropper, and you touch, and replace a particular particle, then at least 1/16 chance of getting caught (1/4 chance that this will be a test particle, and 1/4 chance that you measured the wrong property and then guessed the wrong replacement for the right property). So either the eavesdropper samples at least 100 log n bits and then almost surely gets caught or the eavesdropper samples less than 100 log n bits and then almost surely misses all sqrt(n) bits that was used for the key.

THE REST WAS NOT COVERED THIS SEMESTER

Whirlwind Overview of Shor Algorithm for Factoring (1994): This is viewed as a great result because it shows that you can compute something important quantum mechanically that it seems you can not compute using Newtonian mechanics

Pre-Shor Number theory: If you want to factor a number N, it is sufficient to be able to compute the smallest r such that A^r mod N = 1 for some modest number of random A.

If you look at the set {s: A^s= y_0 mod N}, for some fixed A and y_0, this is an arithmetic progression { (x_0 + r*i} mod N | i=0 1, 2, …}, where

A^(x_0) = y_0 mod N and r is the smallest integer such that A^r = 1 mod N

Aaronson's Thumbtack analogy as a warm up for Shor's algorithm: OK, let me try this. Like many computer scientists, I keep extremely odd hours. You know that famous experiment where they stick people for weeks in a sealed room without clocks or sunlight, and the people gradually shift from a 24-hour day to a 25- or 26- or 28-hour day? Well, that's just ordinary life for me. One day I'll wake up at 9am, the next day at 11am, the day after that at 1pm, etc. Indeed, I'll happily 'loop all the

way around' if no classes or appointments intervene. (I used to do so all the time at Berkeley.)

Now, here's my question: let's say I tell you that I woke up at 5pm this afternoon. From that fact alone, what can you conclude about how long my "day" is: whether I'm on a 25-hour schedule, or a 26.3-hour schedule, or whatever?

The answer, of course, is not much! I mean, it's a pretty safe bet that I'm not on a 24-hour schedule, since otherwise I'd be waking up in the morning, not 5pm. But almost any other schedule — 25 hours, 26 hours, 28 hours, etc. — will necessarily cause me to "loop all around the clock," so that it'd be no surprise to see me get up at 5pm on some particular afternoon.

Now, though, I want you to imagine that my bedroom wall is covered with analog clocks. These are very strange clocks: one of them makes a full revolution every 17 hours, one of them every 26 hours, one of them every 24.7 hours, and so on for just about every number of hours you can imagine. (For simplicity, each clock has only an hour hand, no minute hand.) I also want you to imagine that beneath each clock is a posterboard with a thumbtack in it. When I first moved into my apartment, each thumbtack was in the middle of its respective board. But now, whenever I wake up in the "morning," the first thing I do is to go around my room, and move each thumbtack exactly one inch in the direction that the clock hand above it is pointing.

Now, here's my new question: by examining the thumbtacks in my room, is it possible to figure out what sort of schedule I'm keeping?

I claim that it is possible. As an example, suppose I was keeping a 26-hour day. Then what would happen to the thumbtack below the 24-hour clock? It's not hard to see that it would undergo periodic motion: sure, it would drift around a bit, but after every 12 days it would return to the middle of the board where it had started. One morning I'd move the thumbtack an inch in this direction, another morning an inch in that, but eventually all these movements in different directions would cancel each other out.

On the other hand — again supposing I was keeping a 26-hour day — what would happen to the thumback below the 26-hour clock? Here the answer is different. For as far as the 26-hour clock is concerned, I've been waking up at exactly the same

time each “morning”! Every time I wake up, the 26-hour clock is pointing the same direction as it was the last time I woke up. So I’ll keep moving the thumbtack one more inch in the same direction, until it’s not even on the posterboard at all!

It follows, then, that just by seeing which thumbtack travelled the farthest from its starting point, you could figure out what sort of schedule I was on. In other words, you could infer the “period” of the periodic sequence that is my life.

Shor’s Algorithm to factor N

Action	State
Pick a random A	
	Sum_x |x 0^n>
	Sum_x |x A^x mod N>
Measure the second n bits	
	Sum_l |x_0 + lr> |y_0>
	Where y_0 is random value of A^x, and x_0 + lr is arithmetic progression with period r of values such that A^x=y_0
	Now you would like to find the period r. Note that this is the problem that Aaronson discussed.
Apply Fourier Transform to first n bits	
	Sum_x Sum_l w^((x_0+lr)x) |x>|y_0>

Measure the first n bits. Note that the probability of observing a state |z> with magnitude a + bi is a^ + b^2 (here i is sqrt(-1))

Question: What sort of x’s are you likely to see?

Answer: Note that the x’s here the different clock periods that Aaronson was talking about.

x's that look like ?/r. Probably ? is co-prime with r, so you can get r from x.

Some stuff about Fourier transform, that probably shouldn't be discussed: Let function/vector f(x)=A^x mod N. We liked to find the period of f. A representation of f in terms of periodic functions is a classic problem. Consider the Fourier Transform which transforms a tape in state

f = sum_x |f(x)> |x>

into a state

f' = sum_x |f'(x)> |x>

where f'(x) is defined as

f'(x) = 1/sqrt(N) sum_y w^{xy} f(x)

where w=e^{2 pi i/N} is the primitive Nth root of unity. (Note somewhat similar to Hadamard transform H(x) = sum_y y (-1)^(xy).) . Consider the orthonormal basis {Z_x} where the yth coordinate of Z_x is 1/(sqrt(N) w^(xy)). Note that Z_x is periodic with period x. Then f'(x) can be thought of as the representation of f(x) in the basis {Z_x} consider

1/sqrt(N) sum_y f(x) (w^{xy}) Z_x =

1/N sum_y f(x) =

f(x)

Key fact: If f is periodic with period r, then coefficients f'(r) for basis vector Z_r is big.

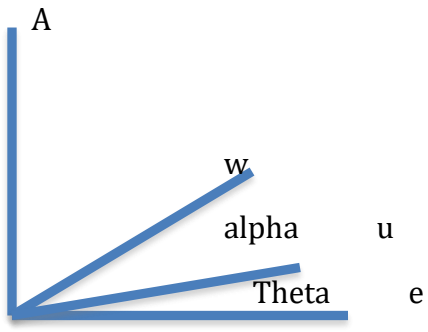
Grover's Algorithm:

It seems unlikely that quantum mechanics allow the possibility of solving NP-complete problems in poly time. But they quantum mechanics does allow for some speed up. Consider the problem of finding a satisfying assignment to a Boolean formula where one is promised that the formula has exactly 1 satisfying assignment A (it is known that one can reduce the general satisfiability problem to this version). Let f be the function from truth assignments to {0,1} defined by f(x) = 1 iff x=A, otherwise f(x)=0.

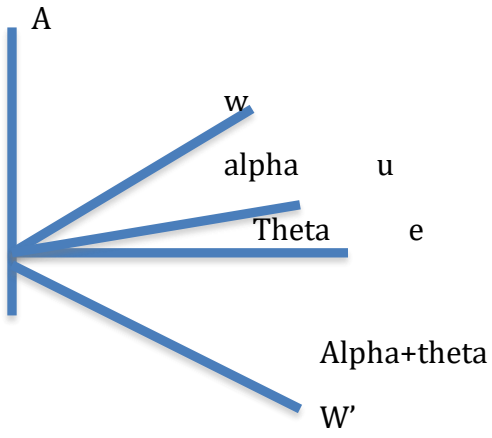
Consider the 2^n dimensional space where there is one basis vector for each possible satisfying assignment. So A is one of the basis vectors. We maintain a vector w in this space. We move w toward A. After a while we will know that w is close to A so that is we measure, we are likely to measure A. Let u be the vector u be the vector ½^{n/2} sum_{x in 2^n} |x> be a uniform superposition of all variable assignments. The vector w is initialized to u.

We repeat the following:

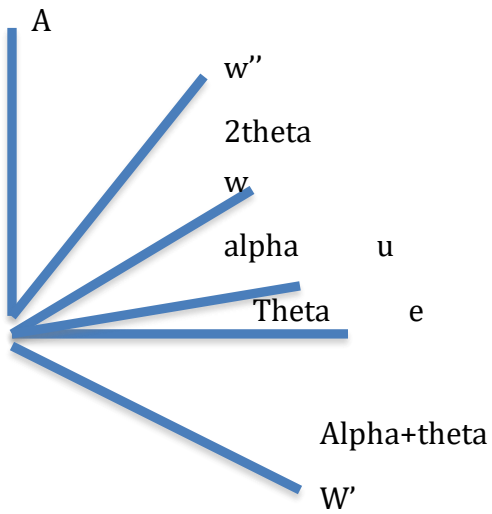
Consider the two dimensional space spanned by the vectors A and u. Let e be the vector orthogonal to A in this space. Note that the algorithm doesn't know the identity of A and e. Let theta be the angle between u and e, and alpha the angle between w and u. See picture



Now rotate w about e to get w'. See picture



Now rotate w' around u to get w'', which will be w for the next iteration. See picture



Note that w is now 2 theta closer to a. Three questions:

1. How many iterations are needed? Since A is a basis vector, Theta = cos^-1 of the inner product of u and A, which is cos^-1 of ½^{n/2} which is about ½^{n/2}. So 2^{n/2} steps move w very close to A.

2. How to rotate about e? We reflect about the hyperplane perpendicular to A. So all coordinates stay the same, except those associated with A are negated. So if w is in state sum b_x |x> it is transformed to state sum b_x (-1)^f(x) |x>. This can accomplished by the reversible code: if f(x)=1 then rotate by pi radians else don't change x

3. How to rotate about u? This one is a little bit trickier. We first change basis, then do a reflection, and then change basis back. We first apply the Hadamard transform to w' to get H(w'). This is the transform that takes u to |0^n>. Then we can reflect around the plane perpendicular to |0^n>. This can can be accomplished by the reversible code: if x=0^n then rotate by pi radians. The Hadamard transform is applied to the result, taking us back to the standard basis.