Quantum Computation

Leonard J. Schulman

Caltech

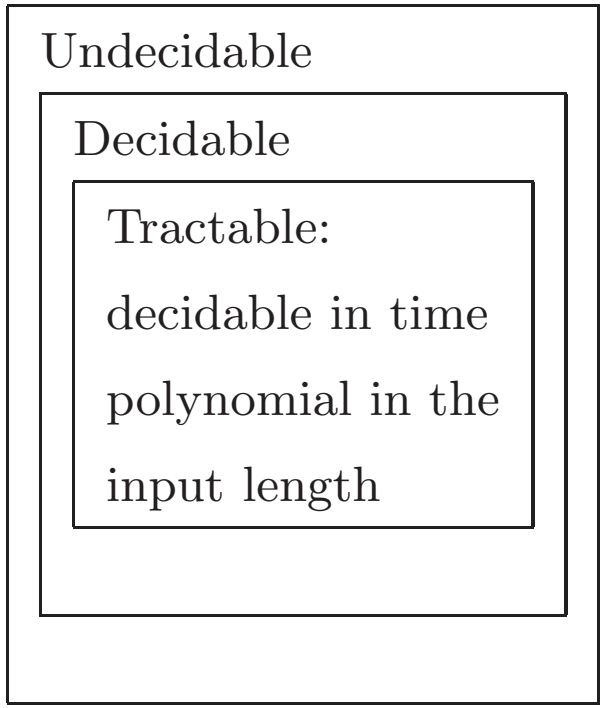
Quantum Computation: what it is, what it isn't.

> Quantum mechanical effects enable efficient solution of classically intractable problems

To appreciate this, we'll review two great lessons of the Twentieth Century: Computational Complexity and Quantum Mechanics.

Logic and Computational Complexity

There are *intrinsic distinctions* in the computational difficulty of mathematical problems.



Quantum Mechanics

A physical system (molecule / cat / computer) is always in a *superposition* of many "definite states". Only interaction with the observer "selects" one of these.

Challenging Computational Problems:

1. "Integrable" physical simulations. Ballistics (Wiener, WWII). Stress analysis. Nuclear physics (Feynman: Manhattan project). Turbulent flow.

2. Cracking secret codes (Turing: Bletchley Park).

3. Logistics and Combinatorial Optimization: Max Flow (Ford, Fulkerson). Resource Allocation, Linear Programming (Dantzig, von Neumann.) Knapsack, Traveling Salesman, Integer Programming.

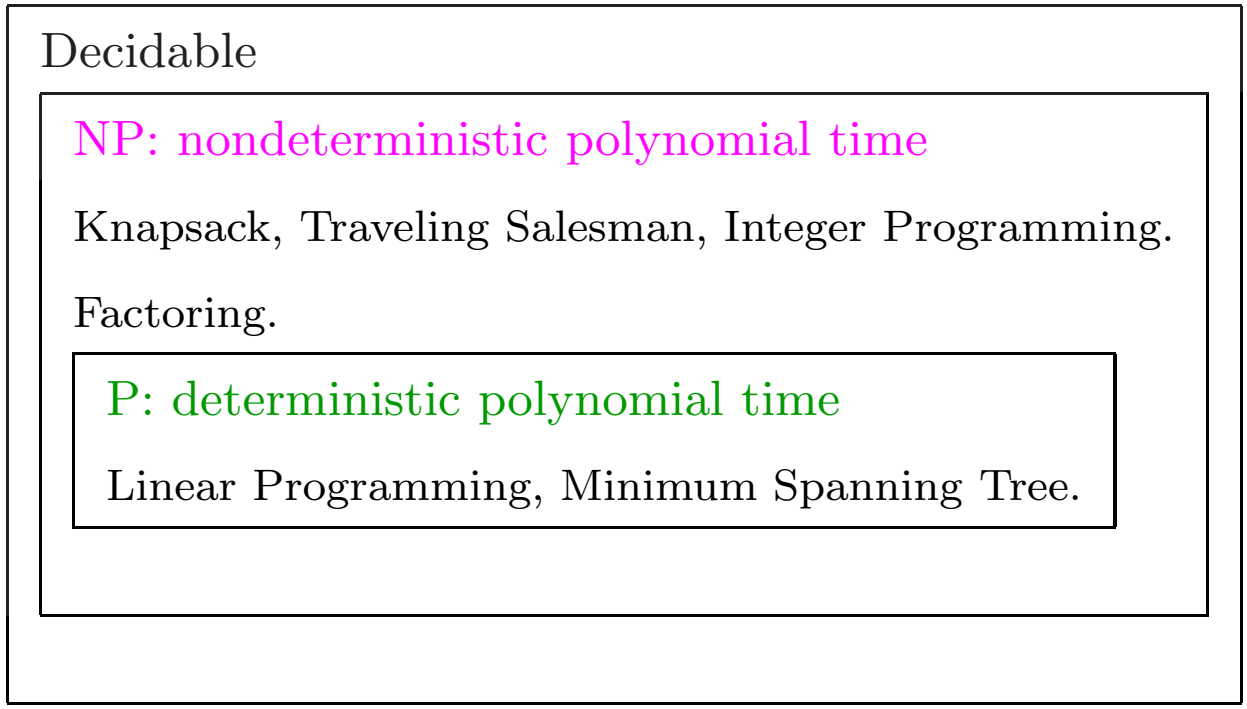
4. Emergent properties of complex systems. Magnets, clouds, Statistical Mechanics ("Ising model"). Highway traffic. Neurons, insect colonies.
   Complex systems are unpredictable $\cong$ they can compute (von Neumann: cellular automata).

5. Simulation of quantum mechanical systems: physical chemistry, particle physics.

Two lessons that were learned in Computer Science:

I. Classification of problems by difficulty. Most importantly: contrast between the difficulty of finding and merely verifying solutions.

(Checking mathematical proofs vs. finding them; calculating the value of a resource allocation vs. finding the best one.)



II. Logic gates and architecture don't matter. *An essential simplifying insight.*

1930's: Logical decidability:

Turing Machine = Church λ-calculus = Post Correspondence Problem = Nondeterministic Turing Machine.

1960's, 1970's: Computational Efficiency:

von Neumann architecture $\cong$ 1-tape Turing Machine $\cong$ 2-tape Turing Machine $\cong$ cellular automaton.

II. Logic gates and architecture don't matter. *An essential simplifying insight.*

1930's: Logical decidability:

Turing Machine = Church λ-calculus = Post Correspondence Problem = Nondeterministic Turing Machine.

1960's, 1970's: Computational Efficiency:

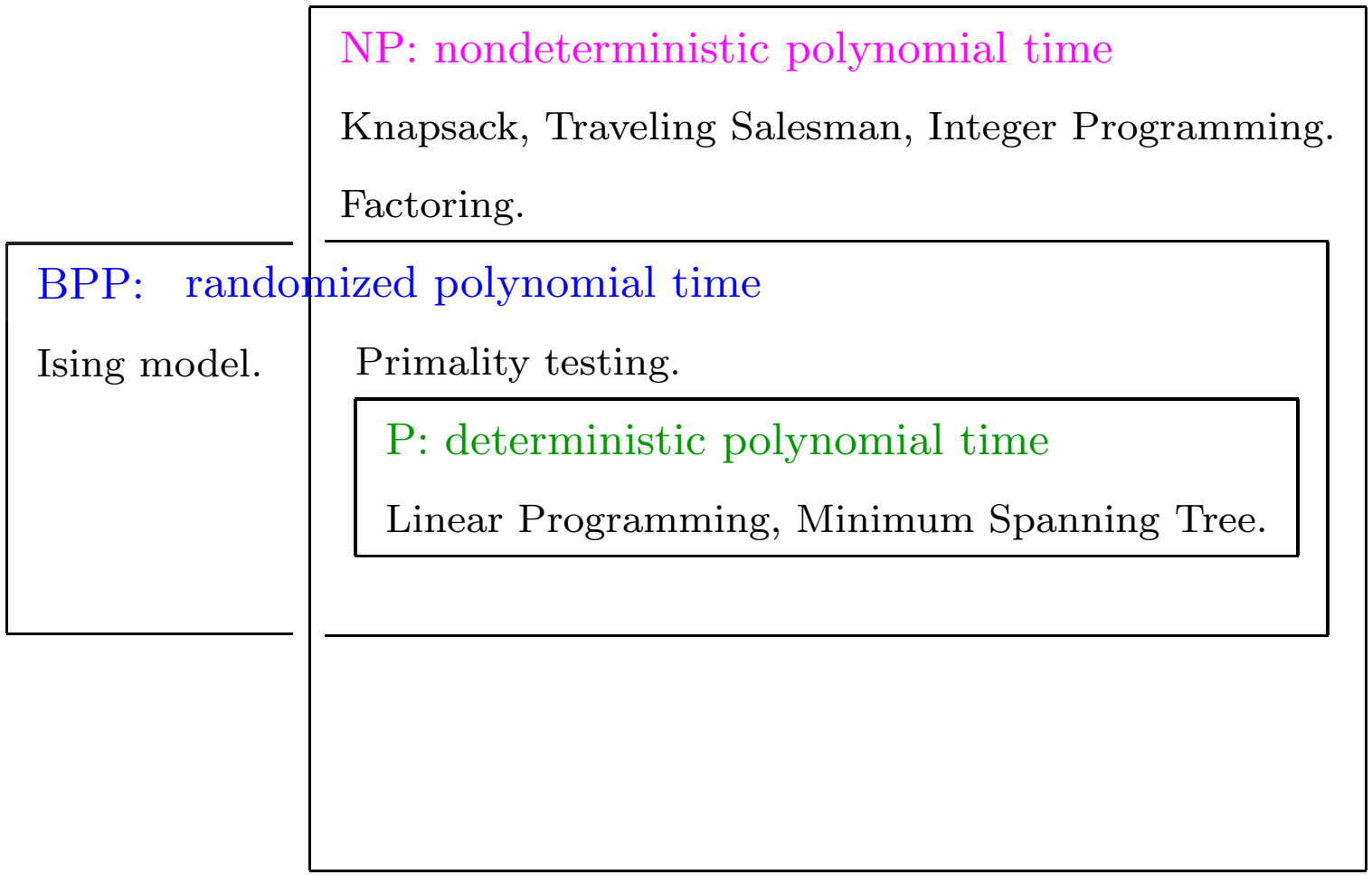
von Neumann architecture $\cong$ 1-tape Turing Machine $\cong$ 2-tape Turing Machine $\cong$ cellular automaton.

... well, not entirely true that "gates don't matter". It helps to have some *totally unreliable gates.*

1950's Metropolis Rosenbluth^2 Teller^2

1970's Rabin

1980's Jerrum Sinclair


NP: nondeterministic polynomial time
Knapsack, Traveling Salesman, Integer Programming.
Factoring.
BPP: randomized polynomial time
Ising model.
Primality testing.
P: deterministic polynomial time
Linear Programming, Minimum Spanning Tree.

We don't know for sure that these complexity classes are really different!

However, every attempt has pointed toward P $\neq$ NP.

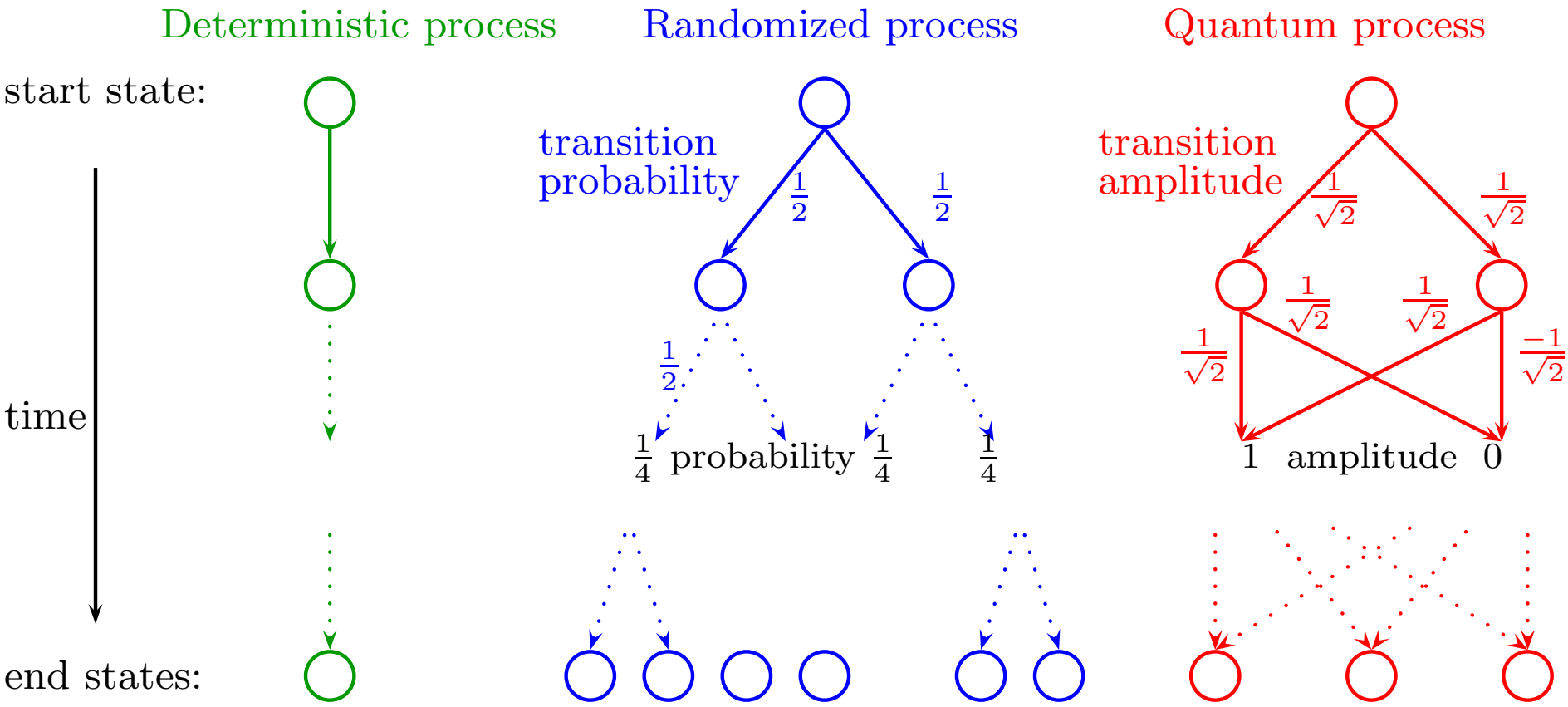
Frustrating if you want efficient algorithms for problems like TSP, Knapsack, Integer Programming, or if you want to put Mathematicians out of work.

But great if you want to hide a secret! P $\neq$ NP offers the possibility of "hiding secrets in plain sight."

Diffie-Hellman: Public key cryptography.
Rivest-Shamir-Adleman (RSA): public key cryptosystem based on the intractability of factoring.

If you *could* factor (or solve similar number-theoretic problems), you could crack all current internet credit card transactions.

Remember "challenging problem 5:" simulating quantum systems.



In each case (deterministic, randomized, quantum) the number of reachable states of the process is $\approx 2^{\text{time}}$.

Feynman 1982: is simulating QM an inherently difficult problem?

Why should simulating quantum mechanics be any more difficult than simulating, say, turbulent systems?

What makes quantum mechanics hard to simulate:

1. *Interference.*

   End-state probability is not predictable from one path. Simulation requires computing entire wave function.

2. *System has m particles $\Rightarrow$ wave function has $\approx 2^m$ amplitudes.*

   Simulating a 300-atom crystal requires writing down 2^{300} complex numbers. But the number of particles in the universe is only $\approx 2^{270}$.

Feynman: Two possibilities:

(a) There's a more clever, classical-polynomial-time (deterministic or randomized), simulation of quantum mechanics.

Feynman: Two possibilities:

(a) There's a more clever, classical-polynomial-time (deterministic or randomized), simulation of quantum mechanics.

Possible, but unlikely.

Feynman: Two possibilities:

(a) There's a more clever, classical-polynomial-time (deterministic or randomized), simulation of quantum mechanics.

Possible, but unlikely.

(b) Quantum mechanics enables efficient solution of classically intractable problems.

Feynman: Two possibilities:

(a) There's a more clever, classical-polynomial-time (deterministic or randomized), simulation of quantum mechanics.

Possible, but unlikely.

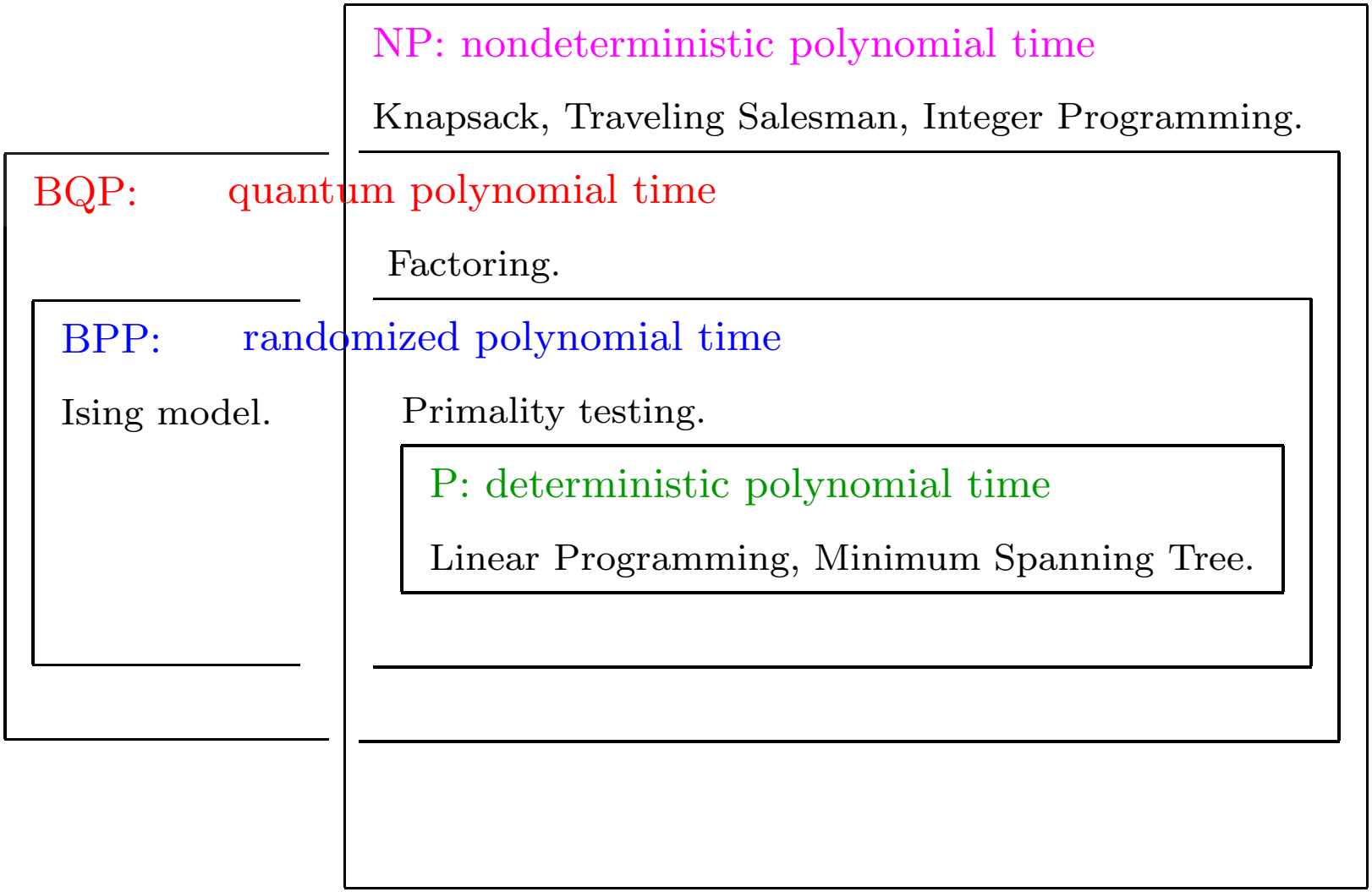
(b) Quantum mechanics enables efficient solution of classically intractable problems.

Logic gates do matter!

Fourier sampling (Bernstein-Vazirani '93)

Fourier sampling simplified (Simon '94)

Factoring (Shor '94)


NP: nondeterministic polynomial time
Knapsack, Traveling Salesman, Integer Programming.
BQP: quantum polynomial time
Factoring.
BPP: randomized polynomial time
Ising model.
Primality testing.
P: deterministic polynomial time
Linear Programming, Minimum Spanning Tree.

Actually... there are two more possibilities.

(c) Quantum mechanics is wrong.

Actually... there are two more possibilities.

(c) Quantum mechanics is wrong.

(d) Quantum mechanics is correct but we just can't engineer these systems.

Next:

1. What is the "architecture" of a quantum computer, and what are some of the leading technologies?
2. Concretely, how does a quantum computer get exponential efficiency gains over classical computers?

1. Architecture: a register of n "qubits"

Each qubit is a particle that has two "basis states," $|0\rangle$ and $|1\rangle$.

Some candidates for qubits:

(a) Qubit = ground $|0\rangle$ vs. excited state $|1\rangle$ of a bound electron.

(b) Qubit = polarization of a photon.

(c) Qubit = ground vs. excited state of an atom trapped in a cavity.

(d) Qubit = polarization of a spin 1/2 nucleus.

The particle can be in any superposition of $|0\rangle$ and $|1\rangle$:

$$\alpha_0 |0\rangle + \alpha_1 |1\rangle \in \mathbb{C}^2 \qquad \text{a 2-dimensional unit vector}$$

A state of the n-qubit computer is a 2^n-dimensional unit vector:

$$w = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle \qquad \text{in the vector space} \qquad \mathbb{C}^2 \otimes ... \otimes \mathbb{C}^2 = \mathbb{C}^{2^n}$$

Logic gates, given by their action on basis vectors:

1. *Not*

$$\begin{aligned} |0\rangle &\rightarrow |1\rangle \\ |1\rangle &\rightarrow |0\rangle \end{aligned}$$

Action on all qubits:

$$\begin{aligned} |x_1 ... 0 ... x_n\rangle &\rightarrow |x_1 ... 1 ... x_n\rangle \\ |x_1 ... 1 ... x_n\rangle &\rightarrow |x_1 ... 0 ... x_n\rangle \end{aligned}$$

2. *Controlled Not*

$$\begin{aligned} |00\rangle &\rightarrow |00\rangle \\ |01\rangle &\rightarrow |01\rangle \\ |10\rangle &\rightarrow |11\rangle \\ |11\rangle &\rightarrow |10\rangle \end{aligned}$$

3. *Hadamard*

$$\begin{aligned} |0\rangle &\rightarrow \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \\ |1\rangle &\rightarrow \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \end{aligned}$$

2. How to get exponential efficiency gains

Key capability of quantum computers: *discovering hidden regularities in very large patterns.*

Key "gate:" poly-time Fourier transform over exponential-size groups.

Most familiar FT application: group $= \mathbb{R}$ or group $= \mathbb{R}/(2\pi)$.
Fourier transform reveals (near)-periodicities of a wave.

Quantum Mechanical Factoring

Want to factor an n-bit number in time poly(n).

Well-known: factoring reduces to order-finding: Given an integer x relatively prime to m, find its *order:* least r such that $x^r \equiv 1 \bmod m$.

A poly-time FT over the group $\mathbb{Z}/(2^n)$ can be implemented on n qubits in time $O(n \log n)$.

$$w \in \mathbb{C}^{(\mathbb{Z}/2^n)}$$

$$\Big\downarrow \text{ Fourier Transform}$$

$$\hat{w} \in \mathbb{C}^{(\mathbb{Z}/2^n)}$$

Measuring the state of the computer after transforming to $\hat{w}$, reveals global structural information about w.

Transforms over $\mathbb{Z}/k$ of some nice waves w:

1. uniform superposition $\longrightarrow$ delta function at the origin
   $w = (\frac{1}{\sqrt{k}}, ..., \frac{1}{\sqrt{k}}) \longrightarrow \hat{w} = (1, 0, ..., 0)$
   equivalently: $\sum_x \frac{1}{\sqrt{k}} |x\rangle \longrightarrow |0...0\rangle$
2. delta function at the origin $\longrightarrow$ uniform superposition
   $w = (1, 0, ..., 0) \longrightarrow \hat{w} = (\frac{1}{\sqrt{k}}, ..., \frac{1}{\sqrt{k}})$
   equivalently: $|0...0\rangle \longrightarrow \sum_x \frac{1}{\sqrt{k}} |x\rangle$
3. uniform superposition on subgroup with period $r \longrightarrow$ uniform superposition on subgroup with period k/r
4. A shift of w changes only phases in $\hat{w}$. $|y\rangle \longrightarrow \sum_x \frac{1}{\sqrt{k}} \omega^{xy \bmod n} |x\rangle$
   (ω is a primitive k'th root of unity.)

Therefore

5. The transform of a shifted subgroup of periodicity r, has uniform *norms* (though varying phases) on the subgroup of periodicity k/r.

Outline of Shor's algorithm to determine the order of x in $\mathbb{Z}/m$:

1. *State preparation*

   Use a FT to transform initial state $|0\rangle$ to (normalizing factor omitted):

   $$\sum_i |i\rangle$$

   and then exponentiate to obtain

   $$\sum_i \left| i, x^i \bmod m \right\rangle$$

   Measure "second register" $x^i \bmod m$. For some uniformly random $1 \leq i \leq \phi(m)$, we're left with the uniform superposition over all j such that $x^j = x^i \bmod m$.

   (Euler totient function $\phi(m) = |\{1 \leq y \leq m, y \text{ rel. prime to } m\}|$.)

   These are all j which differ from i by a multiple of r.

   Hence we have a uniform superposition on the j's in some shift of the subgroup of period r.

2. *Fourier sampling*

   Perform a FT. Obtain a uniform-*norms* superposition on all $\hat{j}$ divisible by $\frac{\phi(m)}{r}$. Sample $\hat{j}$.

Repeat the above two-step process several times.

With high probability the greatest common divisor of the samples $\hat{j}$ is $\frac{\phi(m)}{r}$.

Extract the order r.

How far will this go?

Can we expect exponential gains for all kinds of computational problems?

No. There is almost certainly no efficient quantum algorithm for NP-complete problems. (Bennett-Bernstein-Brassard-Vazirani '97)

Given a "black box" computer program which outputs "yes" on just one of all n-bit binary strings, there is no quantum algorithm to find the "yes" instance in less than time $2^{n/2}$.

(This lower bound was actually matched by a quantum algorithm of Grover.)

Summary

Quantum computation:

What it is:

- Fundamentally different logic gates.
- Speeds up specific kinds of computations.
- A new verification challenge for quantum mechanics.

What it isn't:

- A cure-all for NP-hard problems.

Current research

Quantum algorithms for hard problems.
Quantum information theory.
Implementations: quantum optics, NMR/ESR, doped silicon,...

At Caltech: NSF Institute for Quantum Information. Profs. Doyle, Effros, Kimble, Mabuchi, Preskill, Roukes, Scherer, Schulman.

Coda

Fourier with solemn and profound delight,
Joy born of awe, but kindling momently
To an intense and thrilling ecstasy,
I gaze upon thy glory and grow bright:
As if irradiate with beholden light;
As if the immortal that remains of thee
Attune me to thy spirit's harmony,
Breathing serene resolve and tranquil might,
Revealed appear thy silent thoughts of youth,
As if to consciousness, and all that view
Prophetic, of the heritage of truth
To thy majestic years of manhood due:
Darkness and error fleeing far away,
And the pure mind enthroned in perfect day.

Sir William Rowan Hamilton

reprinted in the American Mathematical Monthly 27 (1920), p.175