Quantum Complexity Theory

Wim van Dam

HP Labs – MSRI – UC Berkeley

SQUINT 3

June 16, 2003

Complexity Theory

Complexity theory investigates what resources (time, space, randomness, etc.) are required to solve certain problems.

Typically, a problem is defined as a language $L \subseteq \{0,1\}^*$ of bit strings. We know how to solve the problem if we can decide between $x \in L$ and $x \notin L$ for every possible $x \in \{0,1\}^*$.

The complexity is expressed as the relation between the length $|x|$ of the input, and the amount of resources required to answer "$x \in L$?"

P: Classical Polynomial Time

The most relevant complexity class is **P**, which contains all problems that can be solved with polynomial time complexity (classically):
$L \in$ **P** if and only if there exists a program that decides $x \in L$? in less than $p(|x|)$ time steps for all x, where p is a polynomial function.

P *contains those problems that we consider 'tractable' or 'efficiently solvable' on a deterministic machine.*
Examples: linear equations, primality testing…

Quantum-P or BQP

"Bounded error, quantum polynomial time": the class of problems that can be solved (with success probability at least 2/3) in polynomial time on a quantum computer.

(The 2/3 is arbitrary: by repeating the algorithm we can amplify the success rate to $1-\varepsilon$.)

BQP is the crucial quantum complexity class.

BPP is bounded error, classical polynomial time.

Question: is **BQP** bigger than **BPP**?

NP and Its Importance

NP stands for **N**ondeterministic **P**olynomial time.

A problem L is in **NP** if and only if for every $x \in L$ there exists a certificate c_x that allows one to efficiently prove that indeed $x \in L$.

Traditional examples are optimization problems:
- Traveling Salesman Problem (TSP): Given n cities and their connections, is there a trajectory that visits all cities in less than T kilometers?
- If so, the trajectory is the certificate of that fact.

Boolean Formulas

A formula $\phi:\{0,1\}^n \rightarrow \{0,1\}$ in n Boolean variables like $\phi(x_1,\ldots,x_n) = (x_1 \neq x_3) \wedge (x_7 \vee \neg x_1) \vee \ldots$

SAT contains all formulas that are satisfiable, $\phi \in$ SAT if and only if $\exists x \in \{0,1\}^n$: $\phi(x)=1$.

Clearly, the SAT problem is in **NP**.
Moreover, SAT is '**NP**-complete': if we have a polytime algorithm to solve SAT, then we are able to solve all **NP** problems in polytime.

The Polynomial Hierarchy

For Boolean functions $\phi(x)$ we can also consider the universal quantifier question: "$\forall x: \phi(x)$?"
This gives the class **co-NP**, which has languages for which there are efficient certificates if $x \notin L$.

By extending the sequence of quantifiers, we get problems like $\exists x'' \forall y' \exists x' \forall y \exists x: \phi(x,x',x'',y,y')$?
The class $\Sigma_k\mathbf{P}$ has problems with k $\exists$-quantifiers.

The "polynomial hierarchy" is the union of all Σ_k:

$$\mathbf{PH} = \bigcup_{k=0,1,2,\ldots} \Sigma_k \mathbf{P}$$

PSPACE

Problems that have polynomial space complexity (but potentially exponential time complexity) are the problems that are in **PSPACE**.

P, **NP**, and the whole polynomial hierarchy are all in **PSPACE** (by reusing the memory).

Embarrassing state-of-the-art: we do not know how to prove that **PSPACE** *is bigger than* **P**. *(We have almost no tools to prove that a problem cannot be solved in polynomial time.)*

Proven vs. Believed Results

For the classical complexity classes we know that:

P $\subseteq$ **NP**$=\Sigma_1$**P** $\subseteq \Sigma_2$**P** $\subseteq \Sigma_3$**P** … $\subseteq$ **PH** $\subseteq$ **PSPACE**

$\subseteq$ **BPP** $\subseteq$

It is generally believed that all these classes are different from each other, except **P** vs. **BPP**.

Actually *proving* one of these difference would be a major scientific advance.
(In the case of **P** vs. **NP**, worth 1,000,000 $.)

Place of BQP

Quantum computers are at least as powerful as classical computers, hence $\mathbf{P} \subseteq \mathbf{BQP}$.

A quantum circuit can be simulated within polynomial space: $\mathbf{BQP} \subseteq \mathbf{PSPACE}$.

Proving $\mathbf{P} \neq \mathbf{BQP}$, implies proving $\mathbf{P} \neq \mathbf{PSPACE}$, which would be a major breakthrough.
(Claims that this has been done are wrong.)

Is BQP Bigger than P?

Factoring, discrete logarithms and solving Pell’s equation are all in **BQP**, and are not known to be in **P**, despite many, many intelligent efforts.

They are known or expected to be in **NP**, but they are unlikely to be **NP**-complete.

The problem of simulating quantum mechanics (≈ predicting quantum circuits) seems unlikely to fit in **P**, but -again- we have no proof of this.

Oracle Results

Problems that concern an outside function (or 'black box') are called 'oracle problems'.

In such 'relativized settings', we often can prove differences between complexity classes like $\mathbf{P}^O \neq \mathbf{NP}^O \neq \mathbf{PSPACE}^O$.

The results of Simon and Shor's period finding give oracles for which $\mathbf{BPP}^O \neq \mathbf{BQP}^O$.

How Big Could BQP Be?

Thus far, everything we know about **BQP** fits in the 2nd level Σ_2**P** of the polynomial hierarchy.

Many researchers consider it unlikely that **BQP** contains all **NP** problems.

Attempts to compute the Permanent problem on a quantum computer are all-but-doomed, because such an algorithm would solve **PH** *problems:* **PH**$\subseteq$**BQP** *has unlikely consequences ("collapses in the hierarchy…")*

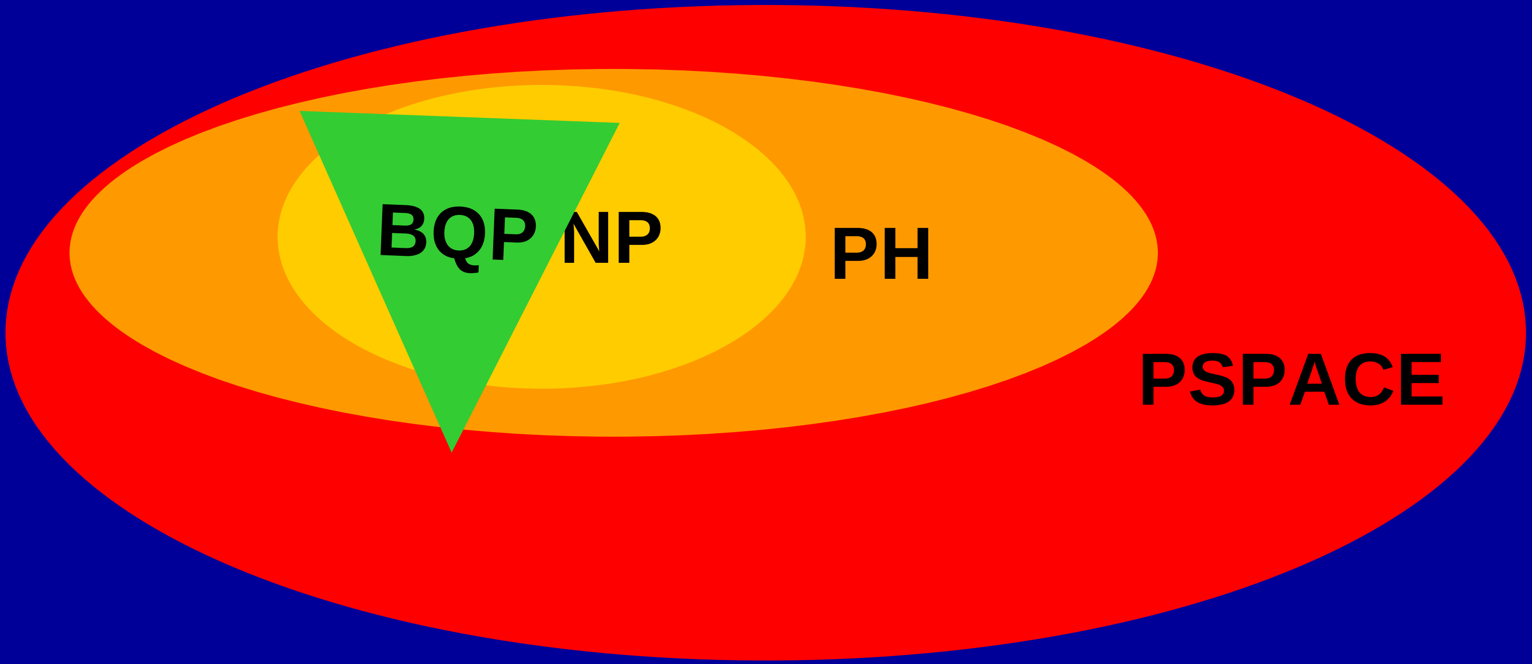
What Could BQP Be?

P=BQP would be a very unexpected result in classical computing.

NP$\subseteq$BQP seems too good to be true.

Likely, **BQP** does not care about the polynomial hierarchy and it contains problems that are somewhat outliers in complexity theory, such as problems in number theory, graph-isomorphism, shortest vector problems, approximate counting…

Perverse Subtlety of BQP



Quantum mechanics seems to favor number theoretic problems over optimization problems?