BPP

Lectures covered the P, NP, and BQP complexity classes. There are many more, but let's briefly introduce one more class which is particularly important.

BPP = bounded-error probabilistic polynomial time

= the class of decision problems which can be solved with a probabilistic classical algorithm with error probability at most $1/3$.

The error probability $1/3$ in the definition is arbitrary, why? As long as it is less than $1/2$, the success probability of the BPP-algorithm can be boosted arbitrarily close to 1. Therefore, BPP rather than P is the class of problems that is usually considered efficiently solvable on a classical computer. BQP is the quantum analogue of BPP and the same probability boosting algorithm can be used there as well.

BPP and the Chernoff bound

Consider a probabilistic algorithm for a decision problem which gives the correct answer with probability $\frac{1}{2} + \varepsilon$ for some fixed $\varepsilon > 0$. Suppose we run this algorithm n times. Then we take the majority vote of the n results as the true output. What is the probability that the majority vote produces the wrong answer?

Chernoff bound

If $X_1, X_2, \ldots, X_n$ are independent, identically distributed random variables s.t.,

$$X_i = \begin{cases} 1 & \text{with probability } \frac{1}{2} + \varepsilon \\ 0 & \text{with probability } \frac{1}{2} - \varepsilon \end{cases},$$

then

$$p\left(\sum_{i=1}^{n} X_i \leq \frac{n}{2}\right) \leq e^{-2\varepsilon^2 n}.$$

Proof:

If $(x_1, x_2, \ldots, x_n)$ is any sequence containing at most $n/2$ ones, then

$$p(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) \leq \left(\frac{1}{2} - \varepsilon\right)^{\frac{n}{2}} \left(\frac{1}{2} + \varepsilon\right)^{\frac{n}{2}}$$

$$= (1 - 4\varepsilon^2)^{\frac{n}{2}} / 2^n.$$

There is surely less than 2^n such sequences, so

$$P\left(\sum_{i=1}^{n} X_i \leq \frac{n}{2}\right) \leq (1-4\varepsilon^2)^{\frac{n}{2}}.$$

Finally since $1-x \leq e^{-x}$, we have the Chernoff bound

$$P\left(\sum_{i=1}^{n} X_i \leq \frac{n}{2}\right) \leq e^{-4\varepsilon^2 n/2} = e^{-2\varepsilon^2 n}.$$

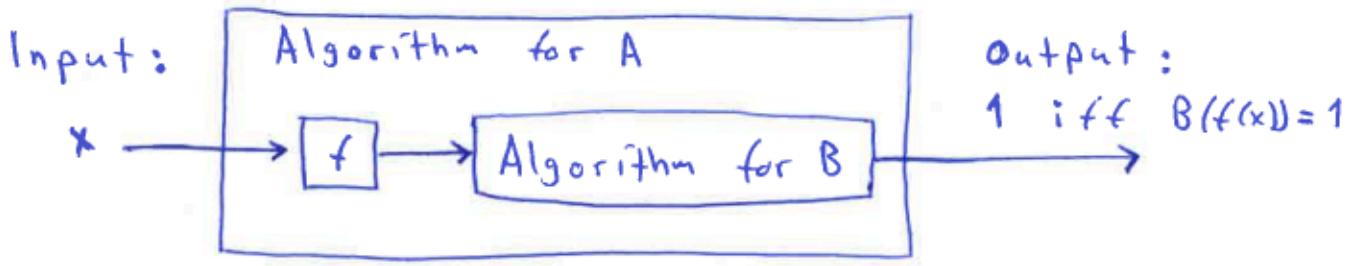
The lesson is that if an algorithm has a failure probability $\varepsilon < \frac{1}{2}$ independent of the problem size, then the failure probability can be suppressed exponentially fast by repetition and taking a majority vote. This is why the failure probabilities in the definitions of BPP and BQP are not very important.

Reductions are important tools in understanding of complexity classes. They can be used to prove that a problem B is "at least as hard" as some other problem A.

Problem A is polynomial-time reducible to problem B if there is a polynomial-time function f s.t. for every instance x of A,

$$A(x) = 1 \text{ if and only if } B(f(x)) = 1.$$

That is, f turns instances of problem A to instances of problem B. Then, we can solve problem A if we have an algorithm for solving B.



Therefore B is at least as hard as A.
Reductions are sometimes denoted $A \leq_p B$.

polynomial-time reduction

NP-hardness and completeness

Amazingly, there exist problems that are at least as hard as any problem in NP. These problems are called NP-hard. A problem L is NP-hard if every NP problem is reducible to it.

NP-hardness: $A \leq_p L$ for any A in NP.

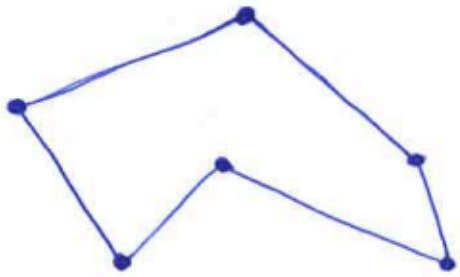
If we had a polynomial-time algorithm for solving L, then we could use it to solve all of NP in polynomial time.

A problem L is called NP-complete if it is both NP-hard and in NP.

Examples of NP-complete problems:

Traveling salesman problem:

Given a list of n cities, the distances between them and a number k, decide if there exist a tour that visits every city that is shorter than k.



Total length $\leq k$

Subset sum:

Given a list of n numbers $a_1, a_2, \ldots, a_n$, and a number T, decide if there is a subset of numbers that sum up to T.

and many more.

It is one of the largest open problems in math to decide whether $P = NP$ or $P \neq NP$. This is the P vs. NP problem.

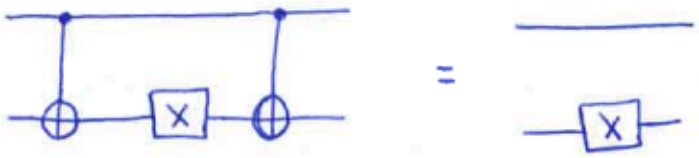
Similar reductions, hardness, and completeness properties also exist for quantum computation (BQP).

Let's now look at some practical examples of quantum circuit calculations.

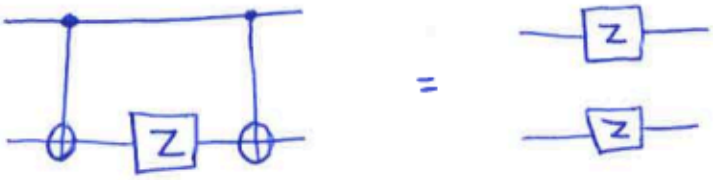
Example:

Prove the following circuit identities.

a) $CX_2C = X_2$



b) $CZ_2C = Z_1Z_2$



c) $CY_2C = Z_1Y_2$

where C is a controlled-NOT with the first qubit as control and the second as target. We'll do these in slightly different ways.

Solution:

a) Since the controls are on the first qubit, we study the action of CX_2C on the states $|0\rangle|\psi\rangle$ and $|1\rangle|\psi\rangle$ where $|\psi\rangle$ is an arbitrary qubit state. Any two qubit state can be written as a linear combination of states of this form.

$$CX_2C\,|0\rangle|\psi\rangle = CX_2|0\rangle|\psi\rangle = C|0\rangle\otimes(X|\psi\rangle)$$

$$= |0\rangle\otimes(X|\psi\rangle) = (\mathbb{1}\otimes X)\,|0\rangle|\psi\rangle$$

$$= X_2\,|0\rangle|\psi\rangle$$

$$C X_2 C\,|1\rangle|\psi\rangle = C X_2\,|1\rangle \otimes (X|\psi\rangle)$$

$$= C\,|1\rangle \otimes (\underbrace{X X}_{=\mathbb{1}}|\psi\rangle) = |1\rangle \otimes (X|\psi\rangle)$$

$$= (\mathbb{1}\otimes X)|1\rangle|\psi\rangle = X_2\,|1\rangle|\psi\rangle .$$

Therefore, $C X_2 C = X_2$ for any two qubit states.

b) This time, let's use matrix algebra.

$$C = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \mathbb{1} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes X = \begin{pmatrix} \mathbb{1} & 0 \\ 0 & X \end{pmatrix}$$

$$Z_2 = \mathbb{1} \otimes Z = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes Z = \begin{pmatrix} Z & 0 \\ 0 & Z \end{pmatrix}$$

Then,

$$C Z_2 C = \begin{pmatrix} \mathbb{1} & 0 \\ 0 & X \end{pmatrix} \begin{pmatrix} Z & 0 \\ 0 & Z \end{pmatrix} \begin{pmatrix} \mathbb{1} & 0 \\ 0 & X \end{pmatrix} = \begin{pmatrix} Z & 0 \\ 0 & \underbrace{XZX}_{=-Z} \end{pmatrix}$$

$$= \begin{pmatrix} Z & 0 \\ 0 & -Z \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes Z = Z \otimes Z$$

$$= Z_1 Z_2 .$$

c) The last one can be done by noticing $Y = iXZ$ and the first two results.

$$C Y_2 C = i\, C X_2 Z_2 C = i\, C X_2 \underbrace{C C}_{=\mathbb{1}} Z_2 C$$

$$= i\, X_2 Z_1 Z_2 = Z_1 (i X_2 Z_2) = Z_1 Y_2$$