The Complexity Classes PostBQP and PP

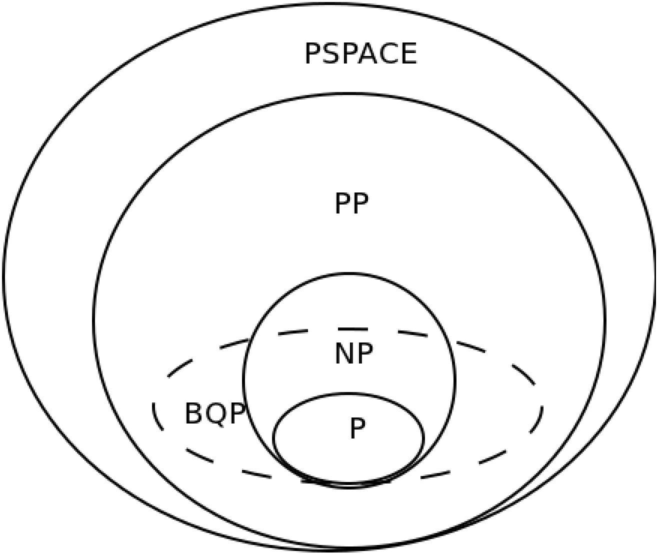
What is the power of postselection in a quantum world?

Alexandre Laplante

QIC710 Project Presentation

PP is the set of problems which can be solved by a probabilistic polynomial time algorithm with probability $> \frac{1}{2}$.
Note that this is not considered to be a feasible set of problems, because the algorithm may only determine an answer with probability $\frac{1}{2} + \frac{1}{2^n}$ where n is the input size. Thus, in order to be sure of an answer, we would need to repeat the algorithm an exponential number of times.

Complexity Theory



PostBQP is the class of problems for which a probabilistic polynomial time quantum algorithm with postselection can determine the solution with probability $\geq \frac{2}{3}$.

PostBQP is the set of languages $L \subseteq \{0,1\}^*$ for which a uniform family of polynomial size quantum circuits $\{G_n\}$ for $n \geq 1$ exists such that when $|\psi\rangle = G_n |x\rangle \otimes |0 \cdots 0\rangle$
(i) The first qubit in $|\psi\rangle$ has > 0 chance of being in the state $|1\rangle$
And when the first qubit is in state $|1\rangle$,
(ii) If $x \in L$ the second qubit in $|\psi\rangle$ has a $\geq \frac{2}{3}$ chance of being in state $|1\rangle$
(iii) If $x \notin L$ the second qubit in $|\psi\rangle$ has a $\leq \frac{1}{3}$ chance of being in state $|1\rangle$

Flip a coin and measure the outcome.
If Tails, kill yourself.
If Heads, do nothing.
What is the probability that you saw heads?

100%! You don't even get asked the question if you saw Tails!

Flip a coin and measure the outcome.
If Tails, kill yourself.
If Heads, do nothing.
What is the probability that you saw heads?

100%! You don't even get asked the question if you saw Tails!

Set-up

Prepare the following state (I'll show how soon):

$$\sqrt{\varepsilon}\,|0\rangle\,|0\rangle\,|1\rangle + \sqrt{1-\varepsilon}\left(\sum_{x\in\{0,1\}^n}\frac{1}{\sqrt{2^n}}\,|1\rangle\,|x\rangle\,|\varphi(x)\rangle\right)$$

Algorithm

Measure the last register.
If you get 0, kill yourself.
Measure the first register.
If you get 1, a solution exists (in the second register)
If you get 0, then either no solution exists, or a solution existed, but you got unlucky and chose $|0\rangle\,|0\rangle\,|1\rangle$ anyway.
Now, we just need to set ε to be small enough that we can be confident that measuring 0 means there was no choice. $\varepsilon = \frac{1}{2^{2n}}$ does the trick.

Set-up

Prepare the following state (I'll show how soon):

$$\sqrt{\varepsilon}\,|0\rangle\,|0\rangle\,|1\rangle + \sqrt{1-\varepsilon}\left(\sum_{x\in\{0,1\}^n}\frac{1}{\sqrt{2^n}}\,|1\rangle\,|x\rangle\,|\varphi(x)\rangle\right)$$

Algorithm

Measure the last register.
If you get 0, kill yourself.
Measure the first register.
If you get 1, a solution exists (in the second register)
If you get 0, then either no solution exists, or a solution existed, but you got unlucky and chose $|0\rangle\,|0\rangle\,|1\rangle$ anyway.
Now, we just need to set ε to be small enough that we can be confident that measuring 0 means there was no choice. $\varepsilon = \frac{1}{2^{2n}}$ does the trick.

Start with

$|0\rangle|0\rangle|0\rangle$

Rotate first qubit

$= \frac{1}{2^n}|0\rangle|0\rangle|0\rangle + \sqrt{1-\frac{1}{2^{2n}}}|1\rangle|0\rangle|0\rangle$

Apply controlled $H^{\otimes n}$ gate to second register

$\frac{1}{2^n}|0\rangle|0\rangle|0\rangle + \sqrt{1-\frac{1}{2^{2n}}}\left(\sum_{x\in\{0,1\}^n}\frac{1}{\sqrt{2^n}}|1\rangle|x\rangle|0\rangle\right)$

Apply the conditional verification algorithm for the NP problem:

$|1\rangle|x\rangle|0\rangle \to |1\rangle|x\rangle|\varphi(x)\rangle$

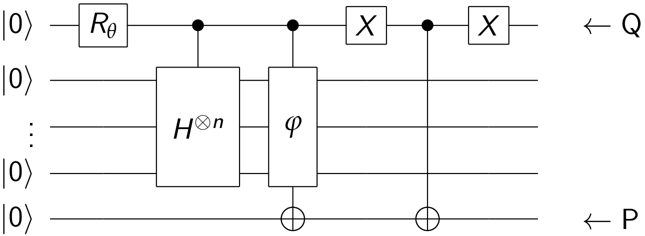
$|0\rangle|x\rangle|0\rangle \to |0\rangle|x\rangle|0\rangle$

$$\frac{1}{2^n}|0\rangle|0\rangle|0\rangle + \frac{\sqrt{1-\frac{1}{2^{2n}}}}{2^{n/2}}\left(\sum_{x\in\{0,1\}^n}|1\rangle|x\rangle|\varphi(x)\rangle\right)$$

Make the third register a 1 if the first register is 0 (X the first register, apply CNOT with the first as control and the last as target, X the first register again)

$$\frac{1}{2^n}|0\rangle|0\rangle|1\rangle + \frac{\sqrt{1-\frac{1}{2^{2n}}}}{2^{n/2}}\left(\sum_{x\in\{0,1\}^n}|1\rangle|x\rangle|\varphi(x)\rangle\right)$$

The Set-up



Another Perspective

Randomly choose an $x \in \{0,1\}^n$.
Run the verification algorithm $\varphi(x)$
If x is not a solution, kill yourself.

Randomly choose an $x \in \{0,1\}^n$.
Run the verification algorithm $\varphi(x)$
If x is not a solution, kill yourself OR do nothing with tiny probability.

If you found a solution, good.
If you didn't, then there's a greater probability that there is no solution than that you survived dispite the existence of a solution.

Randomly choose an $x \in \{0,1\}^n$.
Run the verification algorithm $\varphi(x)$
If x is not a solution, kill yourself.

Randomly choose an $x \in \{0,1\}^n$.
Run the verification algorithm $\varphi(x)$
If x is not a solution, kill yourself OR do nothing with tiny probability.

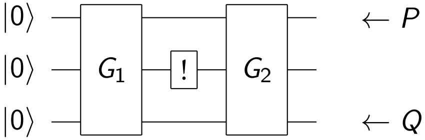
If you found a solution, good.
If you didn't, then there's a greater probability that there is no solution than that you survived dispite the existence of a solution.

Randomly choose an $x \in \{0,1\}^n$.
Run the verification algorithm $\varphi(x)$
If x is not a solution, kill yourself.

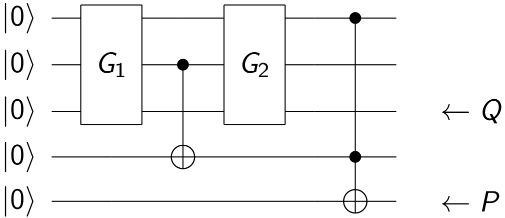
Randomly choose an $x \in \{0,1\}^n$.
Run the verification algorithm $\varphi(x)$
If x is not a solution, kill yourself OR do nothing with tiny probability.

If you found a solution, good.
If you didn't, then there's a greater probability that there is no solution than that you survived dispite the existence of a solution.

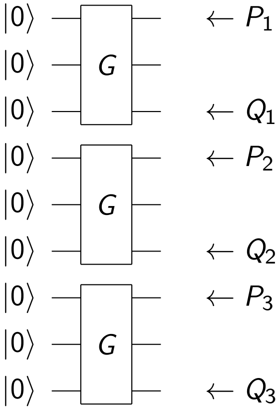
In the definition of PostBQP we've restricted ourselves to postselecting on a single qubit at the very end of the circuit. What if we want to postselect on qubits at intermediate stages in the computation?



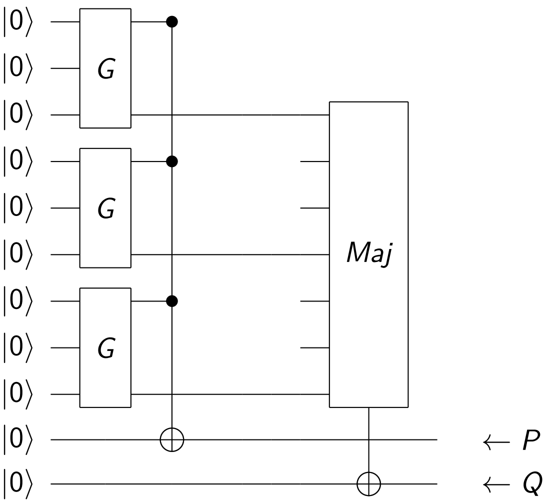
The following circuit takes the AND of the qubit at "!" with the postselected qubit as the new postselected qubit.



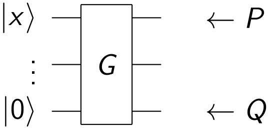
If we want to amplify the success probability from $2/3$ to an arbitrary amount of precision, we can run many copies of the circuit.



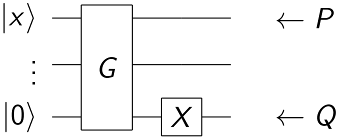
We can use the trick from before to turn this into one run of a PostBQP algorithm with higher success probability.



Is PostBQP = co-PostBQP?
Yes. If we want to know if $x \notin L$, we simply run the algorithm which tests the membership $x \in L$ (call it G) and output the opposite answer. So PostBQP is closed under complement.

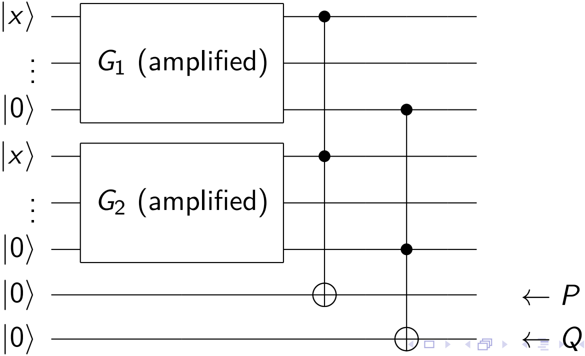


Becomes

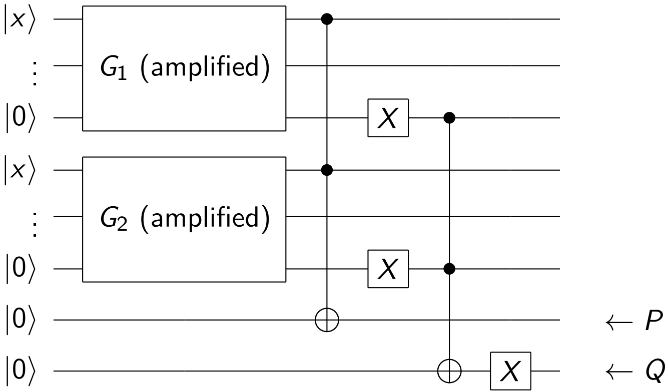


Is PostBQP closed under intersection?

If we want to know if $x \in L_1 \cap L_2$, we amplify the $x \in L_1$ and $x \in L_2$ algorithms (G_1 and G_2 resp.) to each determine with error at most $1/6$. Then (using our trick from earlier to perform a single postselection) we take the "AND" of the two output qubits as our new output qubit. This way we have a PostBQP algorithm to determine if $x \in L_1 \cap L_2$ with error at most $1/3$.



Is PostBQP closed under union?
In exactly the same way, we can design a PostBQP circuit to determine $x \in L_1 \cup L_2$, Using the logical "OR".



Any PostBQP algorithm can be seen as a polynomial number $G(n)$ of Toffoli and Hadamard gates acting on n qubits, followed by a one qubit postselection, and a one qubit measurement.
Applying an algorithm A to an input $|x\rangle$:
$A^G A^{G-1} \cdots A^2 A^1 |x\rangle$

π_1: probability that the output has $P = 1$, $Q = 1$
π_0: probability that the output has $P = 1$, $Q = 0$
Can we make a classical algorithm which accepts with probability $> \frac{1}{2}$ if $\pi_1 > \pi_0$, in polynomial time? (i.e. is PostBQP $\subseteq$ PP ?)

The probability that $P = 1$ and $Q = 1$ is equal to the sum of the squares of the amplitudes of the basis states of $A^G A^{G-1} \cdots A^2 A^1 |x\rangle$ for which $P = 1$ and $Q = 1$. For example: If

$$A^G A^{G-1} \cdots A^2 A^1 |x\rangle = \begin{array}{c} \begin{pmatrix} 1/\sqrt{8} \\ 1/\sqrt{8} \\ 1/\sqrt{8} \\ 1/\sqrt{8} \\ 1/\sqrt{8} \\ 1/\sqrt{8} \\ 1/\sqrt{8} \\ 1/\sqrt{8} \end{pmatrix} \end{array} \begin{array}{ll} 000 & \\ 001 & \\ 010 & \\ 011 & \leftarrow \\ 100 & \\ 101 & \\ 110 & \\ 111 & \leftarrow \end{array}$$

The two positions where the arrows point correspond to $P = 1$ and $Q = 1$ when P and Q are the last 2 qubits.

Let Ψ_ω be the amplitude on the basis state $|\omega\rangle$. In other words, $A^G A^{G-1} \cdots A^2 A^1 |x\rangle = \sum_{\omega=0}^{2^n-1} \Psi_\omega |\omega\rangle$ This is the output of our algorithm.

The probability that the output has $P = 1,\ Q = 1$:

$\pi_1 = \sum_{\omega \in S_1} |\Psi_\omega|^2$

The probability that the output has $P = 1,\ Q = 0$:

$\pi_0 = \sum_{\omega \in S_0} |\Psi_\omega|^2$

Where S_1 is the set of basis states for which $P = 1,\ Q = 1$ and S_0 is the set of basis states for which $P = 1,\ Q = 0$.

Assume (without loss of generality) that Ψ_ω is a real number.
The probability that the output has $P = 1$, $Q = 1$:
$\pi_1 = \sum_{\omega \in S_1} \Psi_\omega \Psi_\omega$
The probability that the output has $P = 1$, $Q = 0$:
$\pi_0 = \sum_{\omega \in S_0} \Psi_\omega \Psi_\omega$

Let's look at what the amplitudes Ψ_ω look like by considering an example:

$$A^G A^{G-1} \cdots A^2 A^1 |x\rangle =$$

$$\begin{pmatrix} A^2_{11} & A^2_{12} & A^2_{13} \\ A^2_{21} & A^2_{22} & A^2_{23} \\ A^2_{31} & A^2_{32} & A^2_{33} \end{pmatrix} \begin{pmatrix} A^1_{11} & A^1_{12} & A^1_{13} \\ A^1_{21} & A^1_{22} & A^1_{23} \\ A^1_{31} & A^1_{32} & A^1_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$= \begin{pmatrix} A^2_{11} & A^2_{12} & A^2_{13} \\ A^2_{21} & A^2_{22} & A^2_{23} \\ A^2_{31} & A^2_{32} & A^2_{33} \end{pmatrix} \begin{pmatrix} \sum_{a=1}^3 A^1_{1a} x_a \\ \sum_{a=1}^3 A^1_{2a} x_a \\ \sum_{a=1}^3 A^1_{3a} x_a \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{b=1}^3 A^2_{1b} \left(\sum_{a=1}^3 A^1_{ba} x_a\right) \\ \sum_{b=1}^3 A^2_{2b} \left(\sum_{a=1}^3 A^1_{ba} x_a\right) \\ \sum_{b=1}^3 A^2_{3b} \left(\sum_{a=1}^3 A^1_{ba} x_a\right) \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{b=1}^{3} \sum_{a=1}^{3} A^2_{1b} A^1_{ba} x_a \\ \sum_{b=1}^{3} \sum_{a=1}^{3} A^2_{2b} A^1_{ba} x_a \\ \sum_{b=1}^{3} \sum_{a=1}^{3} A^2_{3b} A^1_{ba} x_a \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{\forall a,b} A^2_{1b} A^1_{ba} x_a \\ \sum_{\forall a,b} A^2_{2b} A^1_{ba} x_a \\ \sum_{\forall a,b} A^2_{3b} A^1_{ba} x_a \end{pmatrix}$$

So the amplitude of $|\omega\rangle$ is $\sum_{\forall a,b} A^2_{\omega b} A^1_{ba} x_a$

More generally:

$\Psi_\omega = \sum_{\forall \alpha_1, \cdots \alpha_G} A^G_{\omega,\alpha_G} A^{G-1}_{\alpha_G,\alpha_{G-1}} \cdots A^2_{\alpha_3,\alpha_2} A^1_{\alpha_2,\alpha_1} x_{\alpha_1}$

Amplitude for the basis state $|\omega\rangle$ in the output

$$\Psi_\omega = \sum_{\forall \alpha_1, \cdots \alpha_G} A^G_{\omega,\alpha_G} A^{G-1}_{\alpha_G,\alpha_{G-1}} \cdots A^2_{\alpha_3,\alpha_2} A^1_{\alpha_2,\alpha_1} x_{\alpha_1}$$

This is a sum over exponentially many terms.

However, every specific term in this sum is computable in polynomial time. (It is just a product of polynomially many numbers)

Amplitude for the basis state $|\omega\rangle$ in the output

$$\Psi_\omega = \sum_{\forall \alpha_1, \cdots \alpha_G} A^G_{\omega,\alpha_G} A^{G-1}_{\alpha_G,\alpha_{G-1}} \cdots A^2_{\alpha_3,\alpha_2} A^1_{\alpha_2,\alpha_1} x_{\alpha_1}$$

This is a sum over exponentially many terms.

However, every specific term in this sum is computable in polynomial time. (It is just a product of polynomially many numbers)

Amplitude for the basis state $|\omega\rangle$ in the output

$$\Psi_\omega = \sum_{\forall \alpha_1, \cdots \alpha_G} A^G_{\omega,\alpha_G} A^{G-1}_{\alpha_G,\alpha_{G-1}} \cdots A^2_{\alpha_3,\alpha_2} A^1_{\alpha_2,\alpha_1} x_{\alpha_1}$$

This is a sum over exponentially many terms.

However, every specific term in this sum is computable in polynomial time. (It is just a product of polynomially many numbers)

Amplitude for the basis state $|\omega\rangle$ in the output

$\Psi_\omega = \sum_{\forall \alpha_1,\cdots\alpha_G} A^G_{\omega,\alpha_G} A^{G-1}_{\alpha_G,\alpha_{G-1}} \cdots A^2_{\alpha_3,\alpha_2} A^1_{\alpha_2,\alpha_1} x_{\alpha_1}$

Let's call $\psi_{\omega,\alpha} = \left(A^G_{\omega,\alpha_G} A^{G-1}_{\alpha_G,\alpha_{G-1}} \cdots A^2_{\alpha_3,\alpha_2} A^1_{\alpha_2,\alpha_1} x_{\alpha_1}\right)$.
Here α is shorthand for $\alpha_1, \cdots, \alpha_G$. This corresponds to a specific term in the sum. So,
$\Psi_\omega = \sum_\alpha \psi_{\omega,\alpha}$

Now, let's call $X_{\omega,\alpha,\alpha'} = \psi_{\omega,\alpha}\psi_{\omega,\alpha'}$. So,
$\Psi^2_\omega = \left(\sum_\alpha \psi_{\omega,\alpha}\right)\left(\sum_\alpha \psi_{\omega,\alpha}\right) = \sum_{\alpha,\alpha'} \psi_{\omega,\alpha}\psi_{\omega,\alpha'} = \sum_{\alpha,\alpha'} X_{\omega,\alpha,\alpha'}$

Notice again, $X_{\omega,\alpha,\alpha'}$ is computable in polynomial time for any specific ω, $\alpha_1, \cdots \alpha_G$, and $\alpha'_1, \cdots \alpha'_G$.

Amplitude for the basis state $|\omega\rangle$ in the output

$\Psi_\omega = \sum_{\forall \alpha_1, \cdots \alpha_G} A^G_{\omega,\alpha_G} A^{G-1}_{\alpha_G,\alpha_{G-1}} \cdots A^2_{\alpha_3,\alpha_2} A^1_{\alpha_2,\alpha_1} x_{\alpha_1}$

Let's call $\psi_{\omega,\alpha} = \left(A^G_{\omega,\alpha_G} A^{G-1}_{\alpha_G,\alpha_{G-1}} \cdots A^2_{\alpha_3,\alpha_2} A^1_{\alpha_2,\alpha_1} x_{\alpha_1} \right)$.
Here α is shorthand for $\alpha_1, \cdots, \alpha_G$. This corresponds to a specific term in the sum. So,
$\Psi_\omega = \sum_\alpha \psi_{\omega,\alpha}$

Now, let's call $X_{\omega,\alpha,\alpha'} = \psi_{\omega,\alpha}\psi_{\omega,\alpha'}$. So,
$\Psi^2_\omega = \left(\sum_\alpha \psi_{\omega,\alpha} \right) \left(\sum_\alpha \psi_{\omega,\alpha} \right) = \sum_{\alpha,\alpha'} \psi_{\omega,\alpha}\psi_{\omega,\alpha'} = \sum_{\alpha,\alpha'} X_{\omega,\alpha,\alpha'}$

Notice again, $X_{\omega,\alpha,\alpha'}$ is computable in polynomial time for any specific ω, $\alpha_1, \cdots \alpha_G$, and $\alpha'_1, \cdots \alpha'_G$.

Amplitude for the basis state $|\omega\rangle$ in the output

$\Psi_\omega = \sum_{\forall \alpha_1, \cdots \alpha_G} A^G_{\omega,\alpha_G} A^{G-1}_{\alpha_G,\alpha_{G-1}} \cdots A^2_{\alpha_3,\alpha_2} A^1_{\alpha_2,\alpha_1} x_{\alpha_1}$

Let's call $\psi_{\omega,\alpha} = \left(A^G_{\omega,\alpha_G} A^{G-1}_{\alpha_G,\alpha_{G-1}} \cdots A^2_{\alpha_3,\alpha_2} A^1_{\alpha_2,\alpha_1} x_{\alpha_1} \right)$.
Here α is shorthand for $\alpha_1, \cdots, \alpha_G$. This corresponds to a specific term in the sum. So,
$\Psi_\omega = \sum_\alpha \psi_{\omega,\alpha}$

Now, let's call $X_{\omega,\alpha,\alpha'} = \psi_{\omega,\alpha}\psi_{\omega,\alpha'}$. So,

$$\Psi^2_\omega = \left(\sum_\alpha \psi_{\omega,\alpha} \right) \left(\sum_\alpha \psi_{\omega,\alpha} \right) = \sum_{\alpha,\alpha'} \psi_{\omega,\alpha}\psi_{\omega,\alpha'} = \sum_{\alpha,\alpha'} X_{\omega,\alpha,\alpha'}$$

Notice again, $X_{\omega,\alpha,\alpha'}$ is computable in polynomial time for any specific ω, $\alpha_1, \cdots \alpha_G$, and $\alpha'_1, \cdots \alpha'_G$.

Amplitude for the basis state $|\omega\rangle$ in the output

$\Psi_\omega = \sum_{\forall \alpha_1, \cdots \alpha_G} A^G_{\omega,\alpha_G} A^{G-1}_{\alpha_G,\alpha_{G-1}} \cdots A^2_{\alpha_3,\alpha_2} A^1_{\alpha_2,\alpha_1} x_{\alpha_1}$

Let's call $\psi_{\omega,\alpha} = \left(A^G_{\omega,\alpha_G} A^{G-1}_{\alpha_G,\alpha_{G-1}} \cdots A^2_{\alpha_3,\alpha_2} A^1_{\alpha_2,\alpha_1} x_{\alpha_1} \right)$.
Here α is shorthand for $\alpha_1, \cdots, \alpha_G$. This corresponds to a specific term in the sum. So,
$\Psi_\omega = \sum_\alpha \psi_{\omega,\alpha}$

Now, let's call $X_{\omega,\alpha,\alpha'} = \psi_{\omega,\alpha}\psi_{\omega,\alpha'}$. So,
$$\Psi^2_\omega = \left(\sum_\alpha \psi_{\omega,\alpha} \right) \left(\sum_\alpha \psi_{\omega,\alpha} \right) = \sum_{\alpha,\alpha'} \psi_{\omega,\alpha}\psi_{\omega,\alpha'} = \sum_{\alpha,\alpha'} X_{\omega,\alpha,\alpha'}$$

Notice again, $X_{\omega,\alpha,\alpha'}$ is computable in polynomial time for any specific ω, $\alpha_1, \cdots \alpha_G$, and $\alpha'_1, \cdots \alpha'_G$.

We can rewrite our definitions of π_1 and π_0,
The probability that the output has $P = 1$, $Q = 1$:
$$\pi_1 = \sum_{\omega \in S_1} \Psi_\omega \Psi_\omega = \sum_{\omega \in S_1} \sum_{\alpha,\alpha'} X_{\omega,\alpha,\alpha'}$$
The probability that the output has $P = 1$, $Q = 0$:
$$\pi_0 = \sum_{\omega \in S_0} \Psi_\omega \Psi_\omega = \sum_{\omega \in S_0} \sum_{\alpha,\alpha'} X_{\omega,\alpha,\alpha'}$$

Finally, let's construct a classical polynomial time algorithm which accepts with probability $> \frac{1}{2}$ when $\pi_1 > \pi_0$.

Step 1: Choose a random ω, α, and α'.
Step 2: If $\omega \notin S_1 \cup S_2$, accept with probability $\frac{1}{2}$.
Step 3: If $\omega \in S_1$, accept with probability $\frac{1}{2} + \frac{X_{\omega,\alpha,\alpha'}}{2}$.
Step 4: If $\omega \in S_0$, accept with probability $\frac{1}{2} - \frac{X_{\omega,\alpha,\alpha'}}{2}$.

That's it.

Finally, let's construct a classical polynomial time algorithm which accepts with probability $> \frac{1}{2}$ when $\pi_1 > \pi_0$.

Step 1: Choose a random ω, α, and α'.
Step 2: If $\omega \notin S_1 \cup S_2$, accept with probability $\frac{1}{2}$.
Step 3: If $\omega \in S_1$, accept with probability $\frac{1}{2} + \frac{X_{\omega,\alpha,\alpha'}}{2}$.
Step 4: If $\omega \in S_0$, accept with probability $\frac{1}{2} - \frac{X_{\omega,\alpha,\alpha'}}{2}$.

That's it.

Finally, let's construct a classical polynomial time algorithm which accepts with probability $> \frac{1}{2}$ when $\pi_1 > \pi_0$.

Step 1: Choose a random ω, α, and α'.
Step 2: If $\omega \notin S_1 \cup S_2$, accept with probability $\frac{1}{2}$.
Step 3: If $\omega \in S_1$, accept with probability $\frac{1}{2} + \frac{X_{\omega,\alpha,\alpha'}}{2}$.
Step 4: If $\omega \in S_0$, accept with probability $\frac{1}{2} - \frac{X_{\omega,\alpha,\alpha'}}{2}$.

That's it.

What is the probability that this algorithm accepts?

$$= \frac{1}{2} + \frac{\sum_{\omega \in S_1,} \sum_{\alpha,\alpha'} \frac{X_{\omega,\alpha,\alpha'}}{2} - \sum_{\omega \in S_0,} \sum_{\alpha,\alpha'} \frac{X_{\omega,\alpha,\alpha'}}{2}}{\text{number of X's}}$$

From here we see that the probability of accepting is $> \frac{1}{2}$ if and only if

$$\frac{\sum_{\omega \in S_1,} \sum_{\alpha,\alpha'} \frac{X_{\omega,\alpha,\alpha'}}{2} - \sum_{\omega \in S_0,} \sum_{\alpha,\alpha'} \frac{X_{\omega,\alpha,\alpha'}}{2}}{\text{number of X's}} > 0$$

$$\sum_{\omega \in S_1,} \sum_{\alpha,\alpha'} \frac{X_{\omega,\alpha,\alpha'}}{2} - \sum_{\omega \in S_0,} \sum_{\alpha,\alpha'} \frac{X_{\omega,\alpha,\alpha'}}{2} > 0$$

$$\sum_{\omega \in S_1,} \sum_{\alpha,\alpha'} X_{\omega,\alpha,\alpha'} > \sum_{\omega \in S_0,} \sum_{\alpha,\alpha'} X_{\omega,\alpha,\alpha'}$$

$\pi_1 > \pi_0$.

If we have a function $f : \{0,1\}^n \to \{0,1\}$, deciding whether most outputs are 1 is PP-Complete. Here we present a PostBQP algorithm to solve this problem.
If $s = |\{x : f(x) = 1\}|$ is the number of outputs that are 1, and 2^n is the number of possible inputs, we want to know whether $s \geq \frac{2^n}{2}$.
Here we present a PostBQP algorithm which solves this problem.
Note the problem is still PP-Complete if we assume that $s \neq \left\{0, \frac{2^n}{2}, 2^n\right\}$, and we shall.

Step 1: Prepare the state (we'll see how soon)

$$|\psi\rangle = \frac{(2^n - s)|0\rangle + s|1\rangle}{\sqrt{(2^n - s)^2 + s^2}}$$

Step 2: Now prepare the state
$\alpha |0\rangle |\psi\rangle + \beta |1\rangle H |\psi\rangle$
Where α and β are positive real numbers to be determined later.
Note that

$$H|\psi\rangle = \frac{(2^n - s)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + s\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)}{\sqrt{(2^n - s)^2 + s^2}}$$

$$H|\psi\rangle = \frac{(2^n - s)|0\rangle + (2^n - s)|1\rangle + s|0\rangle - s|1\rangle}{\sqrt{2}\sqrt{(2^n - s)^2 + s^2}}$$

$$H|\psi\rangle = \frac{2^n|0\rangle + (2^n - 2s)|1\rangle}{\sqrt{2}\sqrt{(2^n - s)^2 + s^2}}$$

So in total our state is

$$\alpha \left|0\right\rangle \frac{\left(2^n - s\right)\left|0\right\rangle + s\left|1\right\rangle}{\sqrt{\left(2^n - s\right)^2 + s^2}} + \beta \left|1\right\rangle \frac{2^n\left|0\right\rangle + \left(2^n - 2s\right)\left|1\right\rangle}{\sqrt{2}\sqrt{\left(2^n - s\right)^2 + s^2}}$$

Step 3: Postselect on the second qubit being 1. Here we have just measured the second qubit's state as 1, so we erase the 0s and renormalize.

$$\alpha\left|0\right\rangle \frac{s\left|1\right\rangle}{\sqrt{\left(2^n-s\right)^2+s^2}}+\beta\left|1\right\rangle \frac{\left(2^n-2s\right)\left|1\right\rangle}{\sqrt{2}\sqrt{\left(2^n-s\right)^2+s^2}} \text{ (erase 0's)}$$

$$\frac{\alpha s\left|0\right\rangle}{\sqrt{\left(2^n-s\right)^2+s^2}}+\frac{\beta\left(2^n-2s\right)\frac{1}{\sqrt{2}}\left|1\right\rangle}{\sqrt{\left(2^n-s\right)^2+s^2}} \text{ (ignore the second qubit)}$$

$$\left|\varphi_{\beta/\alpha}\right\rangle=\frac{\alpha s\left|0\right\rangle+\beta\left(2^n-2s\right)\frac{1}{\sqrt{2}}\left|1\right\rangle}{\sqrt{\alpha^2s^2+\left(\beta^2/2\right)\left(2^n-2s\right)^2}} \text{ (renormalize)}$$

We call this new state $\left|\varphi_{\beta/\alpha}\right\rangle$.

$$\left|\varphi_{\beta/\alpha}\right\rangle = \frac{\alpha s \left|0\right\rangle + \beta\left(2^n - 2s\right)\frac{1}{\sqrt{2}}\left|1\right\rangle}{\sqrt{\alpha^2 s^2 + \left(\beta^2/2\right)\left(2^n - 2s\right)^2}}$$

If $s < \frac{2^n}{2}$, then $(2^n - 2s) > 0$. This means the amplitude of $|0\rangle$ and the amplitude of $|1\rangle$ are both positive, whatever values we choose for α and β.

If $s > \frac{2^n}{2}$, then $(2^n - 2s) < 0$ for any α and β.

$$\left|\varphi_{\beta/\alpha}\right\rangle = \frac{\alpha s \left|0\right\rangle + \beta \left(2^n - 2s\right) \frac{1}{\sqrt{2}} \left|1\right\rangle}{\sqrt{\alpha^2 s^2 + \left(\beta^2/2\right)\left(2^n - 2s\right)^2}}$$

If $s < \frac{2^n}{2}$, then $(2^n - 2s) > 0$. This means the amplitude of $|0\rangle$ and the amplitude of $|1\rangle$ are both positive, whatever values we choose for α and β.

If $s > \frac{2^n}{2}$, then $(2^n - 2s) < 0$ for any α and β.

$$\left|\varphi_{\beta/\alpha}\right\rangle = \frac{\alpha s\left|0\right\rangle + \beta\left(2^n - 2s\right)\frac{1}{\sqrt{2}}\left|1\right\rangle}{\sqrt{\alpha^2 s^2 + \left(\beta^2/2\right)\left(2^n - 2s\right)^2}}$$

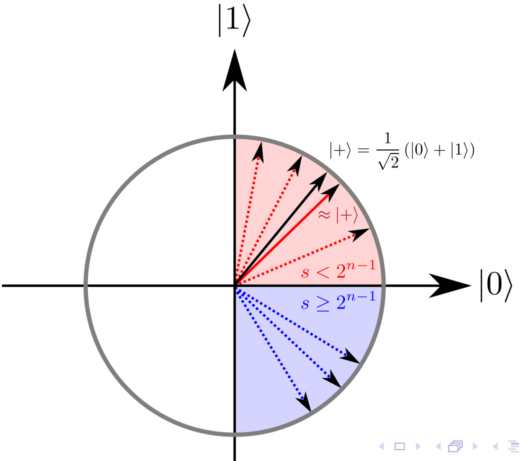
If $s < \frac{2^n}{2}$, then $(2^n - 2s) > 0$. This means the amplitude of $|0\rangle$ and the amplitude of $|1\rangle$ are both positive, whatever values we choose for α and β.

If $s > \frac{2^n}{2}$, then $(2^n - 2s) < 0$ for any α and β.

The Algorithm

$$\left|\varphi_{\beta/\alpha}\right\rangle = \frac{\alpha s \left|0\right\rangle + \beta \left(2^n - 2s\right) \frac{1}{\sqrt{2}} \left|1\right\rangle}{\sqrt{\alpha^2 s^2 + \left(\beta^2/2\right)\left(2^n - 2s\right)^2}}$$

Image taken from http://www.scottaaronson.com/democritus/lec17.html



Prepare $|\varphi_{\beta/\alpha}\rangle$ for many different values of β/α until we find the one where $|\varphi_{\beta/\alpha}\rangle \approx |+\rangle$.

If we ever find it, we reject because $|\varphi_{\beta/\alpha}\rangle$ in the top right quadrant means $s < 2^{n-1}$.

If we never find it, we accept.

Prepare $|\varphi_{\beta/\alpha}\rangle$ for many different values of β/α until we find the one where $|\varphi_{\beta/\alpha}\rangle \approx |+\rangle$.

If we ever find it, we reject because $|\varphi_{\beta/\alpha}\rangle$ in the top right quadrant means $s < 2^{n-1}$.

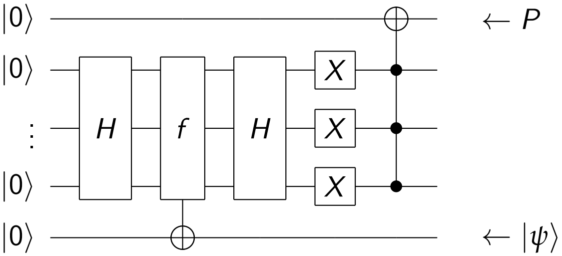
If we never find it, we accept.

Prepare $\left|\varphi_{\beta/\alpha}\right\rangle$ for many different values of β/α until we find the one where $\left|\varphi_{\beta/\alpha}\right\rangle \approx |+\rangle$.

If we ever find it, we reject because $\left|\varphi_{\beta/\alpha}\right\rangle$ in the top right quadrant means $s < 2^{n-1}$.

If we never find it, we accept.

All that is left now is to show how to obtain the initial state
$$|\psi\rangle = \frac{(2^n - s)|0\rangle + s|1\rangle}{\sqrt{(2^n - s)^2 + s^2}}.$$
First prepare the state
$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle$$
Then apply $H^{\otimes n}$ on the first register. And postselect on the first register being in the state $|0^n\rangle$.



This produces the initial state $|\psi\rangle$ for this PostBQP algorithm.

PP $\subseteq$ PostBQP
PP $\supseteq$ PostBQP
PP $=$ PostBQP

We have shown that PostBQP, and thus PP is closed under union, intersection, and complement.

Also, since $\text{PostBQP}^{\text{BQP}} = \text{PostBQP}$, we showed that $\text{PP}^{\text{BQP}} = \text{PP}$.

PP $\subseteq$ PostBQP
PP $\supseteq$ PostBQP
PP $=$ PostBQP

We have shown that PostBQP, and thus PP is closed under union, intersection, and complement.

Also, since $\text{PostBQP}^{\text{BQP}} = \text{PostBQP}$, we showed that $\text{PP}^{\text{BQP}} = \text{PP}$.

PP $\subseteq$ PostBQP
PP $\supseteq$ PostBQP
PP $=$ PostBQP

We have shown that PostBQP, and thus PP is closed under union, intersection, and complement.

Also, since $\text{PostBQP}^{\text{BQP}} = \text{PostBQP}$, we showed that $\text{PP}^{\text{BQP}} = \text{PP}$.

The fact that PP is closed under complementation was first proved in the same paper in which PP was first defined, by John Gill in 1972.

The fact that PP is closed under union and intersection was an open problem for 19 years and was finally proved by Beigel, Reingold, and Spielman in 1991 using something called "threshold polynomials".

The fact that $\mathrm{PP}^{\mathrm{BQP}} = \mathrm{PP}$ was first proved by Fortnow and Rogers in 1999.

The fact that PP is closed under complementation was first proved in the same paper in which PP was first defined, by John Gill in 1972.

The fact that PP is closed under union and intersection was an open problem for 19 years and was finally proved by Beigel, Reingold, and Spielman in 1991 using something called "threshold polynomials".

The fact that $\mathrm{PP}^{\mathrm{BQP}} = \mathrm{PP}$ was first proved by Fortnow and Rogers in 1999.

The fact that PP is closed under complementation was first proved in the same paper in which PP was first defined, by John Gill in 1972.

The fact that PP is closed under union and intersection was an open problem for 19 years and was finally proved by Beigel, Reingold, and Spielman in 1991 using something called "threshold polynomials".

The fact that $\mathrm{PP}^{\mathrm{BQP}} = \mathrm{PP}$ was first proved by Fortnow and Rogers in 1999.

If it were possible to create non-unitary gates, then using $\begin{pmatrix} 2^{-q(n)} & 0 \\ 0 & 1 \end{pmatrix}$ lets you simulate postselection.

If quantum mechanics used a $|\psi|^p$ rule for measurements where $p \neq 2$, we could run the algorithm from earlier which simulates PP. Whenever we need to postselect, if $p < 2$ apply Hadamard gates to $K = 2q(n) / (2 - p)$ ancilla qubits conditioned on $|z\rangle \in S$, this increases the probability mass for each $|z\rangle \in S$ from $|\alpha_z|^p$ to $2^{q(n)} |\alpha_z|^p$ without changing the probability mass of $|z\rangle \notin S$.

If it were possible to create non-unitary gates, then using $\begin{pmatrix} 2^{-q(n)} & 0 \\ 0 & 1 \end{pmatrix}$ lets you simulate postselection.

If quantum mechanics used a $|\psi|^p$ rule for measurements where $p \neq 2$, we could run the algorithm from earlier which simulates PP. Whenever we need to postselect, if $p < 2$ apply Hadamard gates to $K = 2q(n) / (2 - p)$ ancilla qubits conditioned on $|z\rangle \in S$, this increases the probability mass for each $|z\rangle \in S$ from $|\alpha_z|^p$ to $2^{q(n)} |\alpha_z|^p$ without changing the probability mass of $|z\rangle \notin S$.

It's a Win-Win situation. Either:
A) We're completely right about these rules of quantum mechanics

B) We're wrong about quantum mechanics. But who cares? We can solve NP-Complete problems!

It's a Win-Win situation. Either:
A) We're completely right about these rules of quantum mechanics

B) We're wrong about quantum mechanics. But who cares? We can solve NP-Complete problems!

References

- Quantum Computing, Postselection, and Probabilistic Polynomial-Time, Proceedings of the Royal Society A, 461(2063):3473-3482, 2005. quant-ph/0412187.
- PostBQP. (2011, December 13). In Wikipedia, The Free Encyclopedia. Retrieved 22:33, December 21, 2011, from http://en.wikipedia.org/w/index.php?title=PostBQP&oldid=465718
- Leonard M. Adleman, Jonathan DeMarrais, Ming-Deh A. Huang: Quantum Computability. SIAMJ. Comput. 26(5): 1524-1540 (1997)
- J. Watrous. Quantum computational complexity. Encyclopedia of Complexity and System Science, Springer, 2009. Also available as arXiv.org e-Print 0804.3401.
- http://www.scottaaronson.com/democritus/lec17.html