Quantum computing: algorithms and complexity

Assignment 3

Due: 17.05.2023, 23:59

For this assignment you need to recall the precise definitions of a number of complexity classes we considered in the lectures. Except for Problem 5, we will restrict to the case of decision classes. You can find definitions for all of them here: `https://complexityzoo.net/Complexity_Zoo`. You should also consult the course references, in particular the book of Arora and Barak (AB07) and the lecture notes of Ronald de Wolf (W19) and Scott Aaronson (A10). For some of the problems, the solutions can be found in those lecture notes. You are free to consult them if you get stuck, but you should write the proofs in your own words.

Problem 1 (Relations between complexity classes)

In this problem, for all BQP computations you can assume that the YES outcome is given either by measuring a single qubit in the $|0\rangle$ state (or alternatively the $|1\rangle$ state, depending on which labelling we choose), or measuring the entire the output quantum state of the circuit in the $|0^n\rangle$ state and any other output corresponds to a NO outcome. You can choose whichever of these two is more convenient.[1]

A. BQP $\subseteq$ PP. Recall that in the lectures we showed that BQP $\subseteq$ PSPACE by using the Feynman sum-over-paths approach to express the amplitude for a YES outcome $\langle 0^n| C |x\rangle$ as a sum over the amplitudes of exponentially many computational paths. Specifically, if $C = U_T U_{T-1} ... U_1$, where the U_i's are elementary gates, we have:

$$\langle 0^n| C |x\rangle = \sum_{z_1, z_2, ... z_{T-1} \in \{0,1\}^n} \langle 0^n| U_T |z_{T-1}\rangle \langle z_{T-1}| U_{T-1} |z_{T-2}\rangle ... \langle z_1| U_1 |x\rangle . \tag{1}$$

Crucially, each term in the sum can be computed in polynomial time and space (given C and x) and the whole sum can be computed in polynomial space.

We also argued that a similar approach would show that BQP $\subseteq$ P^{PP}. Strengthen this result by showing the containment BQP $\subseteq$ PP. For this, you can assume that the circuit consists only of Hadamard and

[1] And you should convince yourself that they are equivalent. In other words, if the acceptance condition of the circuit is decided by the measurement of a single qubit, one can give an equivalent circuit for which the acceptance condition is decided by measuring the entire output as $|0^n\rangle$. And vice versa.

$CCNOT$ (Toffoli) gates. This is, in fact, without loss of generality, as H and $CCNOT$ can be shown to be universal for quantum computation.[2]

B. $\mathsf{BQPSPACE} = \mathsf{PSPACE}$. We define $\mathsf{BQPSPACE}$ to consist of all decision problems that can be solved by (uniform) quantum circuits using at most a polynomial number of qubits (but potentially exponential time). It's clear that $\mathsf{PSPACE} \subseteq \mathsf{BQPSPACE}$. Show that $\mathsf{BQPSPACE} \subseteq \mathsf{PSPACE}$, therefore $\mathsf{BQPSPACE} = \mathsf{PSPACE}$. In other words, quantum computers cannot offer an exponential advantage in terms of space usage, in general.

Hint: This will again make use of the sum-over-paths approach, though in a different way to how it's been used before. The first thing you should notice is that a $\mathsf{BQPSPACE}$ computation will have at most an exponential number of gates (but will act on a polynomial number of qubits). Denote the circuit for that computation as C. Imagine cutting the circuit at half its depth (or after half of the gates have been performed, either way works). That is, we can express it as $C = C_2 C_1$, where C_1 is the first half of the circuit (the gates up to half the depth of C) and C_2 is the second half. We can then do a sum-over-paths for this partitioning of C, so that:

$$\langle 0^n | \, C \, | 0^n \rangle = \sum_{z \in \{0,1\}^{poly(n)}} \langle 0^n | \, C_2 \, | z \rangle \, \langle z | \, C_1 \, | 0^n \rangle$$

Note that if C_1 and C_2 were just a single gate each (or were depth 1 circuits), the sum could be computed in polynomial space. This is of course not the case, but is a useful observation. Think about what would happen if you continued this process recursively (splitting C_1 and C_2 in half etc).

If you are still unsure about how to proceed, look up Savitch's theorem showing $\mathsf{NPSPACE} = \mathsf{PSPACE}$ (where $\mathsf{NPSPACE}$ is non-deterministic polynomial space, the space analogue of NP).

C. Amplification. Recall that in the definitions of BPP and BQP respectively, we allowed for an error of 1/3. In other words, an input x which is a YES instance is accepted by a polynomial-time randomized (respectively, quantum) algorithm with probability greater than 2/3, while a NO instance is accepted with probability at most 1/3.

Show that the success probability of the algorithm can always be *amplified* so that its maximum error probability is $2^{-p(|x|)}$, for some polynomial p. In other words, YES instances are accepted with probability $1 - 2^{-p(|x|)}$ and NO instances are accepted with probability at most $2^{-p(|x|)}$.

Hint: Taking a majority vote and Chernoff-Hoeffding bounds (see for instance `https://en.wikipedia.org/wiki/Hoeffding%27s_inequality`).

D. $\mathsf{BPP}^{\mathsf{BPP}} = \mathsf{BPP}$. We also argued in the lectures that $\mathsf{P}^{\mathsf{P}} = \mathsf{P}$, since any polynomial-time algorithm that invokes another polynomial-time algorithm as a subroutine leads to an overall polynomial-time algorithm. Show that $\mathsf{BPP}^{\mathsf{BPP}} = \mathsf{BPP}$. You might think that this is result is immediate, but there is a slight subtlety. In $\mathsf{BPP}^{\mathsf{BPP}}$, we are querying an oracle, O, which can be implemented in BPP. Whenever the oracle is queried with some input, x, it will output 0 or 1 *deterministically* depending on whether x is accepted by

[2]Note that this is different from universality in the sense of approximating any unitary. H and $CCNOT$ are clearly not universal in that sense, as they contain only real entries and so one cannot approximate a unitary with complex entries using these gates. Nevertheless, H and $CCNOT$ are universal for quantum computation, in the sense that the output probabilities of any quantum circuit can be reproduced by a circuit composed only of H and $CCNOT$ gates.

the associated BPP computation with probability greater than 2/3 or less than 1/3. Thus, if you were to directly implement O as a BPP computation, you will only obtain the correct 0 or 1 output with probability 2/3. How do you resolve this issue?

Hint: Amplification.

E. $\mathsf{BQP}^{\mathsf{BQP}} = \mathsf{BQP}$. Using ideas from the previous result show that $\mathsf{BQP}^{\mathsf{BQP}} = \mathsf{BQP}$. Here there is another subtlety which does not arise in the classical case. In this case, we model the oracle as a unitary performing the mapping $O|x\rangle|y\rangle = |x\rangle|y \oplus O(x)\rangle$ where $O(x)$ is 0 or 1 depending on whether the associated quantum computation accepts x with probability greater than 2/3 or less than 1/3, respectively (and $|y\rangle$ is one qubit). If we were to directly implement the quantum circuit for the computation that O decides, we only know that the output of that circuit will have high overlap with $|0^n\rangle$ when x is a YES instance and low overlap with $|0^n\rangle$ when it is a NO instance. But note that this does not have the same behavior as O when viewed as a unitary. How do you then implement O as a polynomial-size quantum circuit?

Hint: Uncomputing and amplification.

Problem 2 (Post-selection)

It can be sometimes insightful to consider "unphysical" models of computation and study their computational power in relation to known complexity classes. Towards that end, we will consider probabilistic and quantum computation augmented with *post-selection*. Post-selection is the ability to discard unwanted results and condition the output of the computation on some event (which can occur even with exponentially small, but non-zero, probability).

Let us then define the following classes:

Definition 1 *We say that a promise problem* $\Pi = (YES, NO) \in \mathsf{PostBPP}$ *if there exists a deterministic polynomial-time algorithm* $A(x, r)$ *that outputs two bits* $b_1, b_2 \in \{0, 1\}$*, such that for all* $x \in YES \cup NO$*:*

- *If* $x \in YES$*,*

$$\Pr_{r \leftarrow_U \{0,1\}^{poly(|x|)}}[b_1 = 0 \mid b_2 = 0] \geq 2/3,$$

  where $(b_1, b_2) \leftarrow A(x, r)$*.*

- *If* $x \in NO$*,*

$$\Pr_{r \leftarrow_U \{0,1\}^{poly(|x|)}}[b_1 = 0 \mid b_2 = 0] \leq 1/3,$$

  where $(b_1, b_2) \leftarrow A(x, r)$*.*

In both cases, it should be that $Pr_{r \leftarrow_U \{0,1\}^{poly(|x|)}}[b_2 = 0] > 0$*.*

To give some intuition, our randomized algorithm outputs two bits. We're conditioning (or post-selecting) on the value of the second bit being 0 (which should always happen with non-zero probability) and based on that, the value of the first bit decides whether to accept x.[3] The quantum case is similar:

[3]Note that here $b_1 = 0$ means accept and $b_1 = 1$ means reject.

Definition 2 *We say that a promise problem* $\Pi = (YES, NO) \in \mathsf{PostBQP}$ *if there exists a uniform family of polynomial-size quantum circuits* $\{C_n\}_{n\geq 0}$*, with* C_n *acting on* $m(n) = poly(n)$ *qubits, such that for all* $x \in YES \cup NO$*:*

- *If* $x \in YES$*,*
$$\frac{\|(\langle 00| \otimes I^{\otimes m-2})\, C_n\, |x\rangle\, |0^k\rangle\, \|^2}{\|(I \otimes \langle 0| \otimes I^{\otimes m-2})\, C_n\, |x\rangle\, |0^k\rangle\, \|^2} \geq 2/3.$$
- *If* $x \in NO$*,*
$$\frac{\|(\langle 00| \otimes I^{\otimes m-2})\, C_n\, |x\rangle\, |0^k\rangle\, \|^2}{\|(I \otimes \langle 0| \otimes I^{\otimes m-2})\, C_n\, |x\rangle\, |0^k\rangle\, \|^2} \leq 1/3.$$

where $n = |x|, k = m - n$ *and* $\|(I \otimes \langle 0| \otimes I^{\otimes m-2})\, C_n\, |x\rangle\, |0^k\rangle\, \| > 0$*.*

The fractions in the two conditions simply represent the ratio of the probability of first two qubits being projected onto $|0\rangle$ and the probability that just the second qubit is projected onto $|0\rangle$. Note the use of $\|\cdot\|$ to denote the L_2 norm, since the quantities inside $\|\cdot\|$ are vectors. Just like in the classical case, we're conditioning on the second qubit being $|0\rangle$ (which should occur with non-zero probability) and looking at the outcome of the first to decide acceptance.

It should be clear to you that $\mathsf{PostBPP} \subseteq \mathsf{PostBQP}$ as well as that $\mathsf{BPP} \subseteq \mathsf{PostBPP}$ and $\mathsf{BQP} \subseteq \mathsf{PostBQP}$.

A. In the definitions of PostBPP and PostBQP we are post-selecting on the value of one bit (or qubit). Show that post-selecting on multiple bits (or qubits) does not change the computational power of PostBPP and PostBQP, respectively. It's clear that post-selecting on the value of a single bit (or qubit) is a special case of post-selecting on multiple bits (or qubits). So, clearly, more post-selections can only make the model stronger. What you have to show is that, in fact, post-selections on multiple bits (or qubits) can always be reduced to the case of post-selecting on a single bit (or qubit).

Hint: Logical AND.

B. Show that $\mathsf{NP} \subseteq \mathsf{PostBPP}$. Give a probabilistic algorithm with post-selection that solves an NP-complete problem (for instance SAT). Note that using **A**, you can post-select on as many bits as you like. You should think carefully about how to do this as the "obvious" way of post-selecting on a satisfying assignment doesn't work (when a satisfying assignment doesn't exist, the probability of the post-selected event is 0 and the post-selected event must always have non-zero probability).

This is our first evidence of the "unphysicality" of these classes. With post-selection we would be able to solve NP-complete problems efficiently.

C. Show that $\mathsf{PostBPP} \subseteq \mathsf{PP}$. In fact, one can even show that $\mathsf{PostBPP} \subseteq \Sigma_3$ (where Σ_3 is the 3rd level of the polynomial hierarchy), but here you are only required to show containment in PP. Note that showing containment in P^{PP} is not sufficient, you have to show containment in PP.

Hint: First think about the containment $\mathsf{BPP} \subseteq \mathsf{PP}$ and how post-selection affects it.

D. Show that $\mathsf{PostBQP} \subseteq \mathsf{PP}$.

Hint: Similar to **C**, first consider the proof that $\mathsf{BQP} \subseteq \mathsf{PP}$ from Problem 1 and then see how to extend it to allow for post-selection.

E. Show that $\mathsf{PP} \subseteq \mathsf{PostBQP}$. Together with **D**, this shows that $\mathsf{PostBQP} = \mathsf{PP}$. This is a very surprising result, as $\mathsf{PostBPP} \neq \mathsf{PP}$ unless the polynomial hierarchy collapses. This is yet another indication that quantum computation is more powerful than classical computation—under post-selection quantum computation remains more powerful, unless the polynomial hierarchy collapses.

Hint: To show this, recall that in the lectures we thought about PP as deciding whether a boolean formula ϕ has more than half of its assignments as satisfying or less than half. We saw that we can encode the difference between the number of satisfying and unsatisfying assignments in the amplitude of a certain output state. The problem is that that amplitude is exponentially small, as it is normalized by dividing by the (square root of) total number of assignments. However, with post-selection we can further divide the output probability by the probability of some extremely unlikely event. Leverage this to arrive at an amplitude that is large when more than half of the assignments of ϕ are satisfying, and that is close to 0 otherwise.

Problem 3 (Oracles and query complexity)

A Recall that the decision version of Simon's problem is that we are given oracle access to a function that is promised to be either a Simon function (a 2-to-1 function, f, for which $f(x) = f(y)$ iff. $x = y \oplus s$, for some $s \in \{0,1\}^n$, $s \neq 0^n$) or a 1-to-1 function and we have to accept in the former case and reject in the latter. We showed that relative to this oracle, denoted O, it is the case that $\mathsf{BPP}^{\mathsf{O}} \neq \mathsf{BQP}^{\mathsf{O}}$. One can of course show the same separation with the complement of Simon's problem as well (where the YES and the NO cases are flipped). Use the complement of Simon's problem to show that $\mathsf{BQP}^{\mathsf{O}} \not\subseteq \mathsf{NP}^{\mathsf{O}}$. In other words, show that $coSimon \notin \mathsf{NP}^{\mathsf{O}}$.

Hint: Containment in NP^{O} means there's a deterministic polynomial time algorithm making queries to O, such that, for the YES cases there exists a polynomial-sized witness that makes the algorithm accept. In this case a YES instance is a 1-to-1 function. Fix the witness and examine the queries the algorithms makes in that case. Could those same queries have occurred for a 2-to-1 function?

B In the lectures we saw how to use the polynomial method to lower bound the amount of queries that a quantum algorithm makes to solve the unstructured search problem. We saw that we can re-express that, in the decision-tree model, as approximately computing the OR function over a $N = 2^n$-bit size input. Quantumly this requires $\Omega(\sqrt{N})$ queries to the input (and this is tight as we have a matching upper bound of $O(\sqrt{N})$ with Grover's algorithm).

Use the polynomial method to prove a lower bound of $\Omega(N)$ on the number of quantum queries required for computing the PARITY function. As the name suggests, PARITY computes the PARITY of an N-bit input (PARITY$(X) = 0$, if $|X| \bmod 2 = 0$ and PARITY$(X) = 1$, otherwise).

Hint: Recall the discussion from the end of lecture 10. :)

C Give a quantum algorithm that computes PARITY using $N/2$ queries. That is, for an input $X \in \{0,1\}^N$, give a quantum algorithm that makes $N/2$ queries to this input and which outputs PARITY(X). Recall that quantum queries in this model are queries of the form $|i\rangle \to (-1)^{X_i} |i\rangle$, where $i \in [2^n]$ is the index of the i'th bit of X (denoted X_i).

Hint: Deutsch-Jozsa.

D Consider the function:

$$\text{ZMOD4}(X) = \begin{cases} 1 & \text{if } |X| \mod 4 = 0, \\ 0 & \text{otherwise} \end{cases}$$

The function checks if the Hamming weight of X is a multiple of 4. Using the polynomial method, provide an $\Omega(N)$ lower bound for the quantum query complexity of computing ZMOD4.

Problem 4 (Measurement-based quantum computing (MBQC))

We've seen that general quantum computations can be expressed as circuits comprised of elementary gates that act on qubits. This is called the *circuit model* of quantum computing. An alternative model for representing quantum computations is known as the *measurement-based quantum computing* (MBQC) model. In this model, the idea is to first create a large entangled state. The qubits of this state are then measured adaptively. In other words, the way in which a qubit is measured depends on the measurement outcomes of previous qubits. In this exercise we'll see how this model is equivalent to the usual circuit model.

A First, we will consider a new single-qubit unitary denoted:

$$J(\theta) = HZ(\theta) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & e^{i\theta} \\ 1 & -e^{i\theta} \end{bmatrix}$$

where recall that

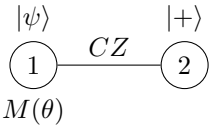
$$Z(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$$

performs a rotation by an angle of θ around the Z axis of the Bloch sphere.

The $J(\theta)$ unitary is universal (for different choices of the angle θ). Specifically, any single-qubit unitary can be written as $U = J(0)J(\theta_1)J(\theta_2)J(\theta_3)$.

Find the J decomposition (i.e. the angles θ_1, θ_2, θ_3) for the cases where U is T, X and Z, respectively.

B With the observation that the $J(\theta)$ unitary is universal, we can now see how single-qubit operations are performed in the MBQC formalism. Suppose we have an input qubit $|\psi\rangle$ on which we wish to apply the $J(\theta)$ operation. We start by initializing another qubit in the $|+\rangle$ state and performing a CZ gate[4] between $|\psi\rangle$ and $|+\rangle$, like in the figure below:

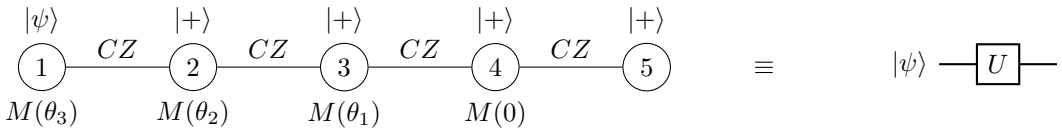


[4]Recall that CZ is symmetric, so it doesn't matter which qubit is the control and which is the target.

We now apply a $Z(-\theta)$ rotation to the qubit 1 and measure it in the $\{|+\rangle, |-\rangle\}$ basis. This is denoted by $M(\theta)$. It is equivalent to measuring in the $\{|+_\theta\rangle, |-_\theta\rangle\}$ basis, where $|\pm_\theta\rangle = Z(\theta)|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle)$. If the measurement outcome was 0 (corresponding to the state having been projected onto $|+\rangle$) we do nothing to qubit 2. If the measurement outcome was 1, we apply a Pauli X to qubit 2. This is referred to as a *correction*.

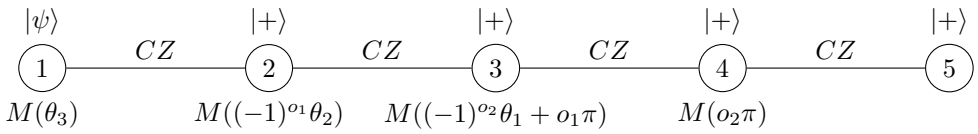
Show that after this process, the state of the second qubit will be $J(\theta)|\psi\rangle$. In other words,

C The previous example illustrated how to apply a single $J(\theta)$ gate to an input qubit in the MBQC formalism. This can be generalized to the application of multiple $J(\theta)$ gates. In particular, consider the MBQC computation below.



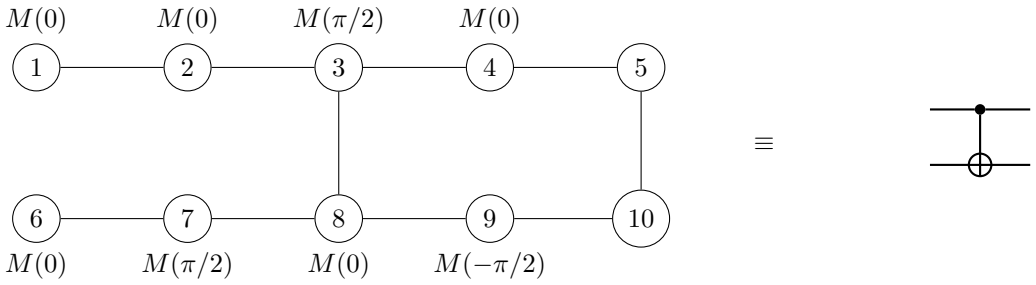
Here, we first initialize a number of qubits in the $|+\rangle$ state and entangle them in a line, together with $|\psi\rangle$ using CZ operations. The resulting state is referred to as a *graph state*. We then measure the qubits in sequence from left to right. Note the reverse order of the θ angles. This is because we want $J(\theta_3)$ to be applied first and $J(0)$ to be last so as to perform the unitary $U = J(0)J(\theta_1)J(\theta_2)J(\theta_3)$. Once again, we need to perform corrections after each measurement. In this case, there will be both X and Z corrections. The corrections on a particular qubit are determined by the measurement outcomes of the previous 2 qubits in the sequence. If the measurement of the previous qubit yielded outcome 1 then we apply an X correction. If the measurement of the qubit before the previous qubit yielded outcome 1 then we apply a Z correction (in other words, corrections are determined by the measurement outcomes of the previous two measured qubits).

Denote the measurement outcomes of qubits 1-4 as $o_1, o_2, o_3, o_4 \in \{0, 1\}$. Show that the corrections can be incorporated into the measurements. In other words, suppose that we perform the following measurements without corrections:



Show that after these measurements and without corrections (i.e. without having to perform on an X or a Z correction on the qubits), the output qubit will be in the state $X^{o_4}Z^{o_3}U|\psi\rangle$, where, as before, $U = J(0)J(\theta_1)J(\theta_2)J(\theta_3)$.

D We now know how to do single-qubit computations in the MBQC model. What about two-qubit computations? For that, we first note that $\{J(\theta), CZ\}$ is a universal set. In other words, any unitary can be expressed in terms of compositions of those two gates. As we've seen, a line graph yields a single-qubit computation. For two-qubit computations we simply have to create 2-dimensional graph states by allowing qubits to entangle on the vertical direction as well. For instance, consider the following graph state and its associated computation (to not overload the figure, we'll suppress the CZ and $|+\rangle$ labels):



In this case the qubits are measured layer by layer going from left to right. We start by measuring the input qubits 1 and 6. We then measure qubits 2 and 7, with the measurements being adapted conditioned on the outcomes of 1 and 6 (in this case, only the measurement of qubit 7 is affected and we would measure with either $M(\pi/2)$ or $M(-\pi/2)$ depending on the outcome of qubit 6).

Show that the given MBQC computation indeed corresponds to performing a $CNOT$ gate. In other words, after all measurements have been performed (and after performing corrections on qubits 5 and 10), the output qubits 5 and 10 will be the $CNOT$ of the input qubits 1 and 6.

Hint: You can ignore the corrections (i.e. assume all measurement outcomes are 0 and there are no corrections). Each horizontal edge and be viewed as the application of a certain $J(\theta)$ gate. A vertical edge is viewed as CZ between the two qubits. With this in mind, show that the sequence of operations being applied is equivalent to $CNOT$.

E Putting everything together, explain how you would translate a general computation from the circuit model to the MBQC model. That is, given as input a quantum circuit C, acting on n qubits, and an input for that circuit $x \in \{0,1\}^n$, explain the steps for performing $C|x\rangle$ in the MBQC model. You can assume that you have a decomposition of all the gates in C in terms of $J(\theta)$ and CZ gates.

Problem 5 (Hardness of circuit sampling via post-selection)

Recall that in the lectures we considered the *weak quantum circuit simulation* task. The task was to exactly sample from the output distribution of a quantum circuit, given as input. As such, we'll rename the task *quantum circuit sampling*. More formally,

Quantum circuit sampling

Input:	Quantum circuit C acting on n qubits and a number $m \leq n$.
Output:	Sample $y \in \{0,1\}^m$ such that $\Pr[y] = \|(\langle y| \otimes I^{n-m})\, C\, |0^n\rangle\, \|^2$.

We showed that if a polynomial-time classical algorithm could succeed at this sampling task, it would lead to a collapse of the polynomial-hierarchy at the 3rd level. In the proof we used Stockmeyer's approximate counting algorithm and the fact that deciding whether the output probability of a general quantum circuit is strictly positive is complete for the class $\mathsf{coC_{=}P}$. Here, you're going to use the tools you've learned in this assignment to give a simpler proof of the hardness of quantum circuit sampling. You will then extend that proof to the case of constant-depth quantum circuits.

A Using post-selection and the ideas developed in Problem 2, show that if a polynomial-time randomized algorithm succeeded at the quantum circuit sampling task it would lead to a collapse of the polynomial hierarchy at the third level. You can assume that it is known that $\mathsf{PostBPP} \subseteq \Sigma_3$ and $\mathsf{PostBQP} = \mathsf{PP}$. Also recall Toda's theorem which says that $\mathsf{PH} \subseteq \mathsf{P^{PP}}$.

Hint: Suppose there's a polynomial-time deterministic algorithm A, such that

$$\Pr_{r \leftarrow_U \{0,1\}^{poly(n)}}[y \leftarrow A(C, x, r)] = \|(\langle y| \otimes I^{n-m})\, C\, |0^n\rangle\, \|^2$$

for all C and $m \leq n$. What does the computational power of the algorithm become under post-selection?

B We now want to extend this argument to the case of constant-depth quantum circuits. To do so, first show that any quantum circuit C can be performed in constant depth using post-selection.

Hint: The application of the circuit C on some state $|x\rangle$ can be viewed as an MBQC computation. Note that creating the graph state can be done in constant depth (why is that?). What would happen if you had no corrections in the MBQC computation?

C Finally, consider the following task:

Shallow quantum circuit sampling

Input:	Constant depth quantum circuit C acting on n qubits and a number $m \leq n$.
Output:	Sample $y \in \{0,1\}^m$ such that $\Pr[y] = \|(\langle y| \otimes I^{n-m})\, C\, |0^n\rangle\, \|^2$.

Using **A** and **B**, show that if a polynomial-time classical algorithm could succeed at this task, the polynomial hierarchy again collapses at the 3rd level.