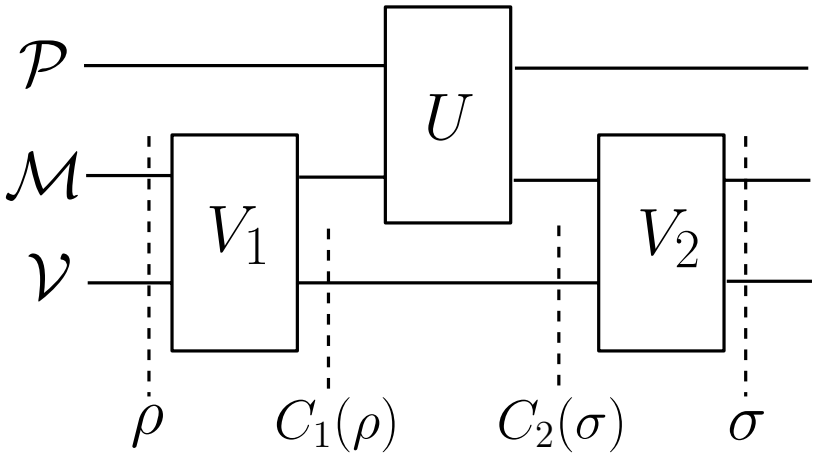
Quantum Reading Group
QIP=PSPACE using multiplicative weights updates [1]

March 29, 2010

We want to show that QIP$(3) \subseteq$ PSPACE since we already know PSPACE $=$ IP $\in$ QIP. One way to show this is to give a PSPACE algorithm which computes the maximum acceptance probability (MAP) of the BQP verifier over all possible provers in an interactive proof.
If language $L \in$ QIP$(3, 1, \epsilon)$, then by definition, if $x \in L$, the MAP will be 1, and if $x \notin L$, the MAP will be $\leq \epsilon$.
The 3 rounds of the quantum interactive proof can be represented by the circuit:



The maximum acceptance probability for the verifier is:
MAP $= F^2(C_1(\rho), C_2(\sigma))$
where F is the fidelity between density matrices.

The above figure leads to a QIP-complete problem: Given two general quantum circuits C_1 and C_2 that operate on mixed states, are there input density matrices ρ and σ such that the two circuits produce almost the same output, or do they differ in output significantly for all inputs, where the difference in outputs is quantified by the square of the fidelity. That is, we want to distinguish between the two cases:

$$\exists \rho, \sigma, \;\; F^2(C_1(\rho), C_2(\sigma)) \geq a$$

$$\forall \rho, \sigma, \;\; F^2(C_1(\rho), C_2(\sigma)) \leq b$$

To show that QIP = PSPACE, we need to give a PSPACE algorithm to distinguish between these two cases for the fidelity between two density matrices. Instead of doing this directly, we will give a PSPACE

algorithm for distinguishing between two cases for the trace distance between two density matrices. We will then be done because the trace distance gives a tight approximation for the fidelity. For two mixed states ρ and σ, the trace distance $||\rho - \sigma||_1$ is the maximum difference in expected outcomes under any PSD measurement; ie. $||\rho - \sigma||_1 = \max_{0 \leq \Pi \leq I} \langle \Pi | (\rho - \sigma) \rangle$.

Our problem is now to distinguish between these two cases: Given circuits C_1 and C_2, is it that

$$\min_{\rho,\sigma} ||C_1(\rho) - C_2(\sigma)||_1 = 0$$

or

$$\min_{\rho,\sigma} ||C_1(\rho) - C_2(\sigma)||_1 \geq 1.9$$

If $D(\mathcal{H})$ is the space of density matrices in our original Hilbert space, another way to write the above is to decide if $\min_{\tau \in D(\mathcal{H}) \otimes D(\mathcal{H})} \max_{0 \leq \Pi \leq I} \langle \Pi | (C_0 \otimes I - I \otimes C_1)(\tau) \rangle$ is 0 or above 1.9. This switch from optimizing over two density matrices ρ and σ to optimizing over a single density matrix τ over a larger space allowed us to write the quantity we wish to approximate as a minimax of an inner product. This should now look familiar; we can now treat this inner product $\langle \Pi | (C_0 \otimes I - I \otimes C_1)(\tau) \rangle$ as the value of a two-player game where player A plays τ and player B plays a measurement Π. We just need to get a good enough estimate of the value of the game in order to distinguish if it is 0 or above 1.9, and we will use multiplicative weights updates (MWU) for this.

Let $C = C_0 \otimes I - I \otimes C_1$. The value of the game is then $\lambda^* = \min\limits_{\tau \in D(\mathcal{H}) \otimes D(\mathcal{H})} \max\limits_{0 \leq \Pi \leq I} \langle \Pi | C\tau \rangle$.

By the minimax theorem,

$$\lambda^* = \min_{\tau \in D(\mathcal{H}) \otimes D(\mathcal{H})} \max_{0 \leq \Pi \leq I} \langle \Pi | C\tau \rangle = \max_{0 \leq \Pi \leq I} \min_{\tau \in D(\mathcal{H}) \otimes D(\mathcal{H})} \langle \Pi | C\tau \rangle$$

The goal of player A is to minimize the inner product, while that of player B is to maximize it. We know, by the minimax theorem, that the value of the game doesn't depend on who plays first. The MWU technique allows us to construct an A who plays first, such that, after a sequence of A-B rounds where A goes first in each round, the value achieved by the MWU algorithm is almost the same as the value achieved in one round where B played first and A played the best response to B. Now if B indeed played his optimal strategy, then the value of this one round where B plays first is just the value of the game, λ^*. Hence, the player A constructed by the MWU algorithm when it plays against an optimal adversary B achieves a value which is approximately the value of the game λ^*. This intuition is made more precise below.

For the multiplicative weights algorithm, we know that for suitably chosen T:

$$\frac{1}{T} \sum_{t=1}^{T} \langle M^{(t)} | C\tau^{(t)} \rangle \leq \frac{1}{1-\epsilon} \left(\frac{1}{T} \sum_{t=1}^{T} \langle M^{(t)} | C\tau^* \rangle + \frac{\log N}{\epsilon} \right)$$

where τ^* is the best non-adaptive strategy A can play given all the measurements $M^{(t)}$ in advance. Clearly $\langle \frac{1}{T} \sum_{t=1}^{T} M^{(t)} | C\tau^* \rangle \leq \max\limits_{0 \leq \Pi \leq I} \min\limits_{\tau \in D(\mathcal{H}) \otimes D(\mathcal{H})} \langle \Pi | C\tau \rangle = \lambda^*$ since $\frac{1}{T} \sum_{t=1}^{T} M^{(t)}$ is just a particular instance of any measurement Π that B could play, and τ^* is optimal against $\frac{1}{T} \sum_{t=1}^{T} M^{(t)}$.

Now, for a fixed value of $\tau^{(t)}$, if B plays the optimal response $M_*^{(t)} = \text{argmax}_{0 \leq \Pi \leq I} \langle \Pi | C\tau^{(t)} \rangle$ to it, then $\lambda^* \leq \langle M_*^{(t)} | C\tau^{(t)} \rangle$ by definition. So we will make the adversary B play the optimal $M_*^{(t)}$ against any strategy $\tau^{(t)}$ played by the MWU algorithm. Finding the optimal $M_*^{(t)}$ for any strategy $\tau^{(t)}$ is easy to describe

- it is simply the projection of $C\tau^{(t)}$ onto the subspace spanned by it's eigenvectors with positive eigenvalues.

Combining these, we have:

$$\lambda^* \leq \frac{1}{T}\sum_{t=1}^{T}\langle M^{(t)}|\tau^{(t)}\rangle \leq \frac{1}{1-\epsilon}\left(\lambda^* + \frac{\log N}{\epsilon}\right)$$

We can estimate the value achieved by the MWU algorithm where the optimal $M_*^{(t)}$ are used by the adversary B against the $\tau^{(t)}$ played by A, and do this in PSPACE. The matrix exponentiations and eigenvector decompositions can all be done without storing their input matrices explicitly, but rather computing each entry as needed on the fly, which can be done using only poly space.

References

[1] Xiaodi Wu. Equilibrium value method for the proof of QIP=PSPACE. Manuscript.