CMSC 858L: Quantum Complexity

Instructor: Daniel Gottesman

Spring 2023

25 QIP = PSPACE

25.1 References

Vidick and Watrous, "Quantum Proofs," arXiv:1610.01664 is again a useful reference for quantum interactive proofs (and also QMA). The specific proof I cover here is from the paper Jain, Ji, Upadhyay, and Watrous, "QIP = PSPACE," arXiv:0907.4737.

25.2 Overview of QIP = PSPACE proof

The basic idea of the proof that QIP = PSPACE is that we can write any quantum interactive proof protocol as a semidefinite program, where the acceptance probability is the function to be maximized in the semidefinite program. There are efficient algorithms to solve semidefinite programs. However, the matrices in this semidefinite program are of exponential size. This does immediately let us conclude that QIP $\subseteq$ EXP, but we need to do more work to show it is in PSPACE.

To do so, the strategy is to write down an explicit algorithm to solve the semidefinite program for a QIP protocol and then to show that the algorithm can be performed with polynomial space. The specific algorithm used here is in a class called multiplicative weights update algorithms.

Before we discuss the algorithm, though, we should discuss semidefinite programs.

25.3 Semidefinite Programming for Quantum Interactive Proofs

A *semidefinite program* is a set of linear equalities or inequalities using positive semidefinite matrices instead of numbers (which would give us a linear program). The goal is to optimize some linear objective function of the matrices. Because density matrices are positive semidefinite matrices, the theory of semidefinite programming is useful in quantum information, particularly when dealing with multiple-round protocols like quantum interactive proofs.

To be more specific, suppose we have a three-round quantum interactive proof. Let ρ_i be the density matrix of the whole system at round i. We could start with ρ_0 as the initial state, but since Alice goes first and her unitary is unknown, it makes more sense to start with ρ_1, the state after Alice performs her initial unitary, which could be any state on AM, but must be correctly initialized on B:

$$\mathrm{Tr}_{AM}\, \rho_1 = |00\ldots0\rangle\langle 00\ldots0|. \tag{1}$$

Now, ρ_2, the state after Bob performs his action, is just related to ρ_1 by Bob's unitary:

$$\rho_2 = (I_A \otimes V_1^B)\rho_1(I_A \otimes V_1^B)^\dagger. \tag{2}$$

Finally, ρ_3 is related to ρ_2 by some unknown unitary on $H_A \otimes H_M$, but must be the same on H_B:

$$\mathrm{Tr}_{AM}\, \rho_3 = \mathrm{Tr}_{AM}\, \rho_2. \tag{3}$$

Finally, Alice is trying to maximize the chance of Bob accepting, which is equal to $\operatorname{Tr}\Pi\rho_3$, where Π is a projector for Bob's final measurement. In this particular case, all the constraints above are equalities rather than inequliaities.

Note that when the protocol is purified, any (ρ_1, ρ_2, ρ_3) that satisfy these equations can be realized through appropriate choice of unitaries by Alice:

Theorem 1. *If ρ and σ on $H_A \otimes H_B$ are any two pure states such that $\operatorname{Tr}_A \rho = \operatorname{Tr}_A \sigma$, then exists U acting on H_A such that $U\sigma U^\dagger = \rho$.*

Proof. This is easily proved via the Schmidt decomposition, which says that there exists bases such that any bipartite pure state

$$|\psi\rangle = \sum_a \sqrt{\lambda_a}|a\rangle \otimes |a\rangle. \tag{4}$$

Then in the Schmidt basis, we can see that $\operatorname{Tr}_A \rho = \operatorname{diag}(\lambda_a)|a\rangle\langle a|$, so when $\operatorname{Tr}_A \rho = \operatorname{Tr}_A \sigma$, that implies that their Schmidt decompositions have the same eigenvalues λ_a. It also implies that the Schmidt bases on B match. (At least, they match exactly if all λ_a are distinct, but if not, we can choose an alternate Schmidt basis for one of them that does match.)

I $\rho = |\psi\rangle\langle\psi|$ and $\sigma = |\phi\rangle\langle\phi|$, then

$$|\psi\rangle = \sum_a \sqrt{\lambda_a}|a\rangle \otimes |a\rangle \tag{5}$$

$$|\phi\rangle = \sum_{a'} \sqrt{\lambda_a}|a'\rangle \otimes |a\rangle. \tag{6}$$

Both $|a\rangle_A$ and $|a'\rangle_A$ are orthonormal bases, so there exists a U acting on H_A alone such that $U|a\rangle = |a'\rangle$. Then

$$U \otimes I|\psi\rangle = \sum_a \sqrt{\lambda_a} U|a\rangle \otimes |a\rangle = |\phi\rangle. \tag{7}$$

□

Semidefinite programs have a *duality* theory. The original semidefinite program above is called a *primal program*, and the dual is the *dual program*. In the dual program, the roles of the constraints vs. objective function, and also the direction of the inequalities are switched.

In particular, if the primal semidefinite program is:

- Variables X, positive semidefinite
- Maximize $\langle C, X\rangle$
- Subject to $\Psi(X) \leq D$

then the dual program is

- Variables Y, positive semidefinite
- Minimize $\langle D, Y\rangle$
- Subject to $\Psi^*(Y) \geq C$

Here, $\Psi^*(Y)$ is the adjoint, $\langle \Psi(X), Y\rangle = \langle X, \Psi^*(Y)\rangle$. The inner product is $\langle C, X\rangle = \operatorname{Tr} C^\dagger X$.

The value of this duality is that the minimal value of the dual program is always greater than or equal to the maximum value of the primal program. This is improved to having the minimal dual value exactly equal to the maximum primal value if some additional condition is met, for instance if there is an interior feasible point for the dual program (i.e., a solution for which the inequalities in the dual program are strictly satisfied).

25.4 The semidefinite program we will use

For this proof, we first make one more simplification to the protocol. Note that in a 3-round protocol, Alice sends to Bob, then Bob sends back to Alice, and finally Alice sends to Bob again. When we apply the transformation from last time, it can't make this shorter, but it can further simplify it if we alter it slightly. In the transformation, we could start by Alice sending Bob the state σ (on H_B) he would have after round 2 instead of round 1 (when Bob has sent a transmission back to Alice). Bob then sends a single bit to Alice saying whether to go forward or backward. Alice goes next, and she either finishes the protocol by sending the last transmission to Bob or sends back to Bob the state she would have received from him for round 2. When they reverse, Bob can then reverse his round 2 unitary V_1^B to get his initial state and a message received from Alice; he only needs to check that his initial state has been returned to the state $|00\ldots0\rangle$.

Basically, Bob sends his bit to Alice who sends him more qubits, giving him either the state ρ_0 or ρ_1 on $H_M \otimes H_B$, and then Bob projects on either P_0 or P_1 and accepts only if that projection succeeds. Let

$$Q = \frac{1}{2}\begin{pmatrix} P_0 & 0 \\ 0 & P_1 \end{pmatrix}, \quad \tilde{X} = \begin{pmatrix} \rho_0 & 0 \\ 0 & \rho_1 \end{pmatrix}. \tag{8}$$

The additional degree of freedom (representing the value of the random bit Bob sends to Alice) we will call C.

Since Alice cannot have the initial state she sends Bob depend on the random bit, we have that $\operatorname{Tr}_M \rho_0 = \operatorname{Tr}_M \rho_1 = \sigma$. Therefore, $\operatorname{Tr}_M \tilde{X} = I_C \otimes \sigma$. The chance of acceptance is $\operatorname{Tr}(Q\tilde{X})$. Alice can choose σ and $\tilde{X}$ arbitrarily subject to these constraints, but Q is given by the protocol. Thus, this is a semidefinite programming problem.

To get this program in the form used in the proof, we first relax the constraint $\operatorname{Tr}\tilde{X} = I_C \otimes \sigma$ to $\operatorname{Tr}\tilde{X} \leq I_C \otimes \sigma$. Since the only constraint on σ is that it is a density matrix (and thus has trace 1), the relaxation doesn't increase the achievable value of $\operatorname{Tr}(Q\tilde{X})$, since if the inequallity is not tight, we can scale X up linearly to make it tight; this increases $\operatorname{Tr}(Q\tilde{X})$, so the maximum value of $\operatorname{Tr}(QX)$ is always achieved when the inequality is at its maximum value. We can also relax the constraint that $\operatorname{Tr}\sigma = 1$ to $\operatorname{Tr}\sigma \leq 1$.

The next step is to let $\Phi(D) = \operatorname{Tr}_M(Q^{-1/2}DQ^{-1/2})$ and $\Phi^*(D) = Q^{-1/2}(D \otimes I_M)Q^{-1/2})$. This is an adjoint in the sense that $\operatorname{Tr}(\Phi(D)^\dagger E) = \operatorname{Tr}(D^\dagger \Phi^*(E))$. Here, $\Phi : H_M \otimes H_C \otimes H_B \to H_C \otimes H_B$ and $\Phi^* : H_C \otimes H_B \to H_M \otimes H_C \otimes H_B$.

Next we write the program in terms of the variable $X = Q\tilde{X}$ instead of $\tilde{X}$. We then get the following *primal program:*

- Variables X, σ positive semidefinite.
- Maximize $\operatorname{Tr}(X)$
- Subject to $\Phi(X) \leq I_C \otimes \sigma$ and $\operatorname{Tr}\sigma \leq 1$.

To compute its dual, let us put in the form above. The variables are (X, σ), we have $C = I$,

$$\Psi(X, \sigma) = (\Phi(X) - I \otimes \sigma, \operatorname{Tr}\sigma), \tag{9}$$

and $D = (0, 1)$. Then the dual variables should be (Y, y), Y positive semidefinite, $y \geq 0$ a number, and

$$\langle (X, \sigma), \Psi^*((Y, y))\rangle = \operatorname{Tr}[([\Phi(X)]^\dagger - I_C \otimes \sigma)Y] + y \operatorname{Tr}\sigma \tag{10}$$

This gives us

$$\Psi^*(Y, y) = (\Phi^*(Y), yI - \operatorname{Tr}_C Y). \tag{11}$$

We then get the constraints $\Phi^*(Y) \geq I$ and $yI \geq \operatorname{Tr}_C Y$, and we want to minimize y. Now, the second constraint $yI \geq \operatorname{Tr}_C Y$ is achieved whenever y is at least the maximum eigenvalue of $\operatorname{Tr}_C Y$, so to minimize y, we should minimize the maximum eigenvalue of $\operatorname{Tr}_C Y$. This gives us the following *dual program*:

- Variables Y positive semidefinite.
- Minimize $\|\operatorname{Tr}_C(Y)\|_{sup}$
- Subject to $\Phi^*(Y) \geq I$

25.5 Algorithm to solve the semidefinite program

To solve this program, we use the following algorithm:

1. Initialize $W_0 = I$, $Z_0 = I$, constants γ, ϵ, δ, T.
2. We always have $\rho_i = W_i / \operatorname{Tr} W_i$, $\xi_i = Z_i / \operatorname{Tr} Z_i$.
3. Run for T steps:
   (a) Let Π_i be the projection onto the positive eigenspaces of
   $$\Phi(\rho_i) - \gamma I_C \otimes \xi_i. \tag{12}$$
   (b) Let $\beta_i = \operatorname{Tr}(\Pi_i \Phi(\rho_i))$. If $\beta_i \leq \epsilon$, then accept and end.
   (c) Otherwise, let
   $$W_{i+1} = \exp\left(-\epsilon\delta \sum_{j=0}^{i} \Phi^*(\Pi_j/\beta_j)\right) \tag{13}$$
   $$Z_{i+1} = \exp\left(\epsilon\delta \sum_{j=0}^{i} \operatorname{Tr}_C(\Pi_j/\beta_j)\right) \tag{14}$$
4. If it hasn't accepted after T steps, reject.

Note that

$$\operatorname{Tr} W_{i+1} = \exp\left(-\epsilon\delta \sum_{j=0}^{i} \Phi^*(\Pi_j/\beta_j) - \epsilon\delta\Phi^\dagger(\Pi_{i+1}/\beta_{i+1})\right) \leq \operatorname{Tr}[W_i \exp(-\epsilon\delta\Phi^*(\Pi_{i+1}/\beta_{i+1}))] \tag{15}$$

by the Golden-Thompson inequality $\operatorname{Tr}\exp(A+B) \leq \operatorname{Tr}(\exp A \exp B)$. Similarly,

$$\operatorname{Tr} Z_{i+1} \leq \operatorname{Tr}[Z_i \exp(\epsilon\delta \operatorname{Tr}_A(\Pi_j/\beta_j))]. \tag{16}$$

This is the sense in which this is a multiplicative weights update algorithm.

At all times i, let

$$X = \frac{\rho_i}{\gamma + 2\operatorname{Tr}(\Pi_i\Phi(\rho_i))} \tag{17}$$

$$\sigma = \frac{\gamma\xi_i + 2\operatorname{Tr}_C(\Pi_i\Phi(\rho_i)\Pi_i)}{\gamma + 2\operatorname{Tr}(\Pi_i\Phi(\rho_i))}. \tag{18}$$

Then one can show that (X, σ) is a primal solution. If we accept, it is true that

$$\operatorname{Tr} X = \frac{1}{\gamma + 2\operatorname{Tr}(\Pi_i\Phi(\rho_i))} \geq \frac{1}{\gamma + 2\epsilon}, \tag{19}$$

which is close to $1/\gamma$ for small ϵ. In particular, for an appropriate choice of parameters, this is higher than the maximum probability of acceptance for a "no" instance, meaning that it must be a "yes" instance. Thus, the algorithm accepts only for "yes" instances.

If we reject, then

$$Y = \frac{1+2\epsilon}{T} \sum_{j=0}^{T-1} \Pi_j/\beta_j \tag{20}$$

is a dual solution. If we take the same combination for a smaller T, it is not a feasible dual solution, but with each iteration of the algorithm we add to Y, making it larger and larger. After an appropriate number of steps, we have $\Phi^\dagger(Y) \geq I$ and it is a valid solution. In this case, we find that

$$\mathrm{Tr}_C\, Y \leq (1 + 2\epsilon)(\exp(\epsilon)/\gamma + \epsilon^2/4), \tag{21}$$

which again is close to $1/\gamma$ for small ϵ. In particular, this definitely means that the optimal dual solution is below 1, which is the probability of acceptance in the "yes" instance, meaning this is a "no" instance. The algorithm rejects only for "no" instances, so it is correct in all cases.

25.6 Performing the algorithm in PSPACE

The goal will be to approximate each matrix element of any matrix that appears in the algorithm using polynomial space. Then traces, determinants, and inner products of those matrices can be calculated with exponential sums, which can be done in PSPACE.

Note that we can find matrix elements of Q in polynomial space using the usual methods of simulating a quantum algorithm with polynomial space. We can also find eigenvalues of Q by computing the characteristic polynomial (using the determinant) and using numerical methods to approximate its roots. We can solve the linear equations to find an eigenvector corresponding to a particular eigenvalue of Q. This lets us approximate matrix elements of $Q^{-1/2}$. This in turn lets us compute matrix elements of $\Phi(D)$ and $\Phi^*(D)$ when needed.

Finally, we approximate exponentials by taking a finite number of terms in the sum, giving us a composition of usual matrix operations, all of which can be done in PSPACE.

To complete the proof, we need to analyze how accurately all this can be done. This requires some work, but the bottom line is that these quantities can be computed very precisely, meaning the algorithm still gives the correct answer when implemented in this way.

Thus, there is a PSPACE algorithm that can correctly decide any language in QIP.