CS 880: Quantum Information Processing 11/4/2010

Lecture 25: Error Correction

Instructor: Dieter van Melkebeek Scribe: Brian Nixon

In this lecture, we complete our discussion of interactive proof systems and begin talking about error correcting codes. We gradually build a system that will secure a single qubit from errors and prove its correctness. Next lecture we will discuss a different method of single qubit error correction.

1 Quantum Interactive Proof Systems, continued

Recall from last lecture that classically a language L has an interactive proof system (IPS) iff $L \in$ PSPACE. In the quantum setting, both the verifier and the prover gain addition power so it is not immediately clear how the class of languages that have a quantum interactive proof system (QIPS) compares. We know that we do not lose power as the verifier can simply force the prover to adopt a classical protocol by making an observation after each step. So we at least have QIPS for all of PSPACE. A recent result proved that we get nothing more than PSPACE by demonstrating that any QIPS can be reduced to a similar type as we used in our graph isomorphism protocol last lecture. In this standard form there are 4 steps:

1. Prover sends a register X of qubits to Verifier,
2. Verifier picks $b \in \{0, 1\}$ uniformly at random and sends it to Prover,
3. Prover sends register Y to Verifier,
4. Verifier decides whether to accept.

Given this form we want to know the probability that the verifier accepts. We can capture this behavior with a semi-definite program. Semi-definite programs (SDPs) are similar to linear programs – they optimize a linear objective function over the reals under linear inequality constraints; in addition they can use constaints that require that certain variables from a semi-definite matrix. They solution to SDPs can be approximated to within ϵ in time polynomial in the size of the program and $1/\epsilon$.

In our case the objective function can be written as

$$\frac{1}{2}\sum_{i=0}^{1} \Pr(\text{Verifier accepts}|b = i) = \frac{1}{2}\operatorname{Tr}(\pi_0\rho_0) + \frac{1}{2}\operatorname{Tr}(\pi_1\rho_1),$$

where π_i denotes the projection onto the accepting subspace in the case $b = i$, and ρ_i denotes the density operator related to the combination of X and Y for $b = i$. The constraints are the following.

- ρ_0 and ρ_1 are density operators. This can be expressed in and SDP program by stipulating that ρ_0 and ρ_1 are positive semi-definite matrices, and the linear equality that they have trace 1.

- X is independent of b. This can be expressed by the linear equalities that tracing out Y from ρ_0 and ρ_1 yields the same matrix: $\mathrm{Tr}_Y(\rho_0) = \mathrm{Tr}_Y(\rho_1)$.

The resulting SDP has exponential size but has enough structure such that it can be solved in polynomial space.

2 Multiple Prover IPS

How does the model change if we allow multiple provers? Such provers are allowed to collaborate prior to the start of the protocol but are not allowed to communicate once it has begun (or else there would effectively be a single prover). In the classical system, the verifier would benefit from such a situation by being able to check for consistency between the various provers. This way teh verifier can effectively force the provers to act in a nonadaptive manner where their anwers do not depend on the questions asked before. It is known that classically a 2-prover IPS exists exactly for languages in NEXP (nondeterministic exponential time), and that more than 2 provers doesn't buy any more. With quantum systems, the effect of multiple provers is an open question. The key difficulty here is that provers can share entangled qubits prior to the start of the protocol and cannot be viewed as completely disconnected.

3 Bloch Sphere and Pauli operators

Our discussion of quantum error correcting codes will be limited to codes that can correct errors on a single qubit. Before starting that discussion, we first review an interesting geometric representation of single qubit systems and operations.

Recall that an operation on a single qubit can be represented by a 2-by-2 matrix. Let $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$, and $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. These are the Pauli operators. Each of them are Hermitian and unitary so induce a legal quantum operation on a single qubit. The following properties of the operators are useful and you should confirm them for yourself.

Exercise 1. *1. $XY = iZ$, $YZ = iX$, $ZX = iY$.*

2. $HXH = Z$ where H is the Hadamard matrix.

3. $\{I, X, Y, Z\}$ form a basis for all 2×2 matrices.

As operations on qubits, X performs a bit flip, Z performs a phase flip, and Y corresponds to a combination of the two. The second item in Exercise 1 shows that Z corresponds to a bit flip in the Hadamard basis. The third item implies that all single qubit density operators can be decomposed into $\rho = \alpha I + \beta X + \gamma Y + \delta Z$. We can easily show that $\alpha = 1/2$ in this case as $\mathrm{Tr}(\rho) = 1$, $\mathrm{Tr}(I) = 2$, $\mathrm{Tr}(X) = \mathrm{Tr}(Y) = \mathrm{Tr}(Z) = 0$, and the trace is a linear operator on matrices. This allows us to write any single qubit density operator as

$$\rho = \frac{1}{2}I + \frac{1}{2}(c_x X + c_y Y + c_z Z). \tag{1}$$

A pure state of a single qubit can be written uniquely as $|\psi\rangle = \cos(\theta)|0\rangle + e^{i\phi}\sin(\theta)|1\rangle$ up to a global phase shift where $0 \leq \phi < 2\pi$ and $0 \leq \theta \leq \frac{\pi}{2}$. Calculating the corresponding density

operator $\rho = |\psi\rangle\langle\psi|$ yields

$$\rho = \begin{pmatrix} \cos^2(\theta) & e^{-i\phi}\sin(\theta)\cos(\theta) \\ e^{i\phi}\sin(\theta)\cos(\theta) & \sin^2(\theta) \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1+\cos(2\theta) & e^{-i\phi}\sin(2\theta) \\ e^{i\phi}\sin(2\theta) & 1-\cos(2\theta) \end{pmatrix}. \tag{2}$$

Equating (1) and (2) yields

$$\begin{cases} c_x &= \sin(2\theta)\cos(\phi) \\ c_y &= \sin(2\theta)\sin(\phi) \\ c_z &= \cos(2\theta) \end{cases}$$

These are the polar coordinates of a point on a three-dimensional sphere with radius 1 centered at the origin. The representation is known as the Bloch sphere.

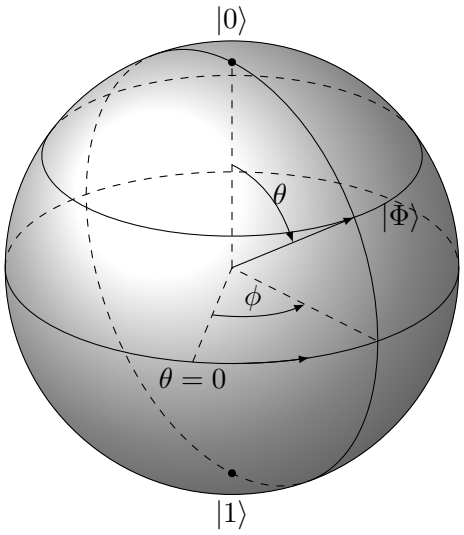

Figure 1: Bloch Sphere

Every single qubit density operator corresponds uniquely to a point on the Bloch sphere. What about mixed states? These are convex combinations of pure states so will yield a point on the interior of the sphere.

What is the effect of the Pauli operators on the Bloch sphere? Applying Z adds π to ϕ, which amounts to a 180° rotation around the z-axis (or if you prefer, a reflection through the z-axis). A little calculation will reveal that X performs a rotation of 180° around the x-axis, and Y performs a rotation of 180° around the y-axis. Any curiosity you might have been holding towards why the Pauli operators had their specific names should now be resolved. In general, any unitary operation on $|\psi\rangle$ corresponds to a rotation (not necessarily of 180°) around some axis in the Bloch sphere.

Exercise 2. *Show the Hadamard matrix performs a reflection through the plane including the y and $(x+z)$ axes.*

We note that $|\langle\phi_1|\phi_2\rangle|^2 = \mathrm{Tr}(\rho_1\rho_2) = \frac{1}{2} + \frac{1}{2}(c_{x_1}c_{x_2} + c_{y_1}c_{y_2} + c_{z_1}c_{z_2})$. In particular, orthogonal states map to antipodal points on the Bloch sphere.

4 Error Correction

Error correction seems considerably harder in the quantum setting than in the classical setting, and we will only discuss error correction on a single qubit. For comparison, correction on a single bit is easy - consider the code that repeats each bit thrice. To recover any bit, simply take the majority view of any triple (taking the view that errors are unlikely to affect more than one bit out of three).

Why is error correction for a qubit so much harder than the simple code we gave for the classical bit? There are three main reasons. First, a qubit has a continuum of possibilities where a bit's value lies in a discrete set. Second, it is easy to copy a bit but in the quantum setting we have the no cloning rule restricting us. Third, if we do a measurement in the course of detecting and correcting errors, that will collapse the state and may destroy information.

In the face of these apparent difficulties we first restrict our question farther to simply correcting for the possibility of a bit flip error, noting that this corresponds to the only possible error in the classical environment.

Taking our cue from the classical code, let us consider a system of three qubits that we bind together by applying CNOT's to get $\alpha|000\rangle + \beta|111\rangle$. See the first half of Figure 2. There are four possible output states after a single bit flip error, namely $\alpha|000\rangle + \beta|111\rangle$, $\alpha|100\rangle + \beta|011\rangle$, $\alpha|010\rangle + \beta|101\rangle$, and $\alpha|001\rangle + \beta|110\rangle$. Fortunately, these exist in orthogonal subspaces so we can separate them perfectly. In fact, we can figure out the error and correct if efficiently, as indicated in the second half of Figure 2. Consider the effect of reapplying the CNOT gates. If bit flips occurred or the bit flips occurred on the extra qubits, this results in the first qubit returning to $\alpha|0\rangle + \beta|1\rangle$. We can correct the remaining error case by adding a CNOT gate that modifies the first qubit and is controlled by the other two. After this procedure the other two qubits take on a pure value of either $|0\rangle$ or $|1\rangle$. In the field of error correction these extra two bits are called "syndromes" and they tell us exactly what error occurred:

00 for no error

01 for a bit flip on the third qubit

10 for a bit flip on the second qubit

11 for a bit flip on the first qubit.

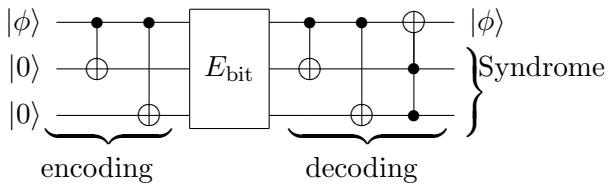

Figure 2: Bit flip correction

It is important to note that if there is a phase flip during the error stage, it appears in the final state $|\phi\rangle$ on the first qubit. You should verify this for yourself.

To correct a phase flip we can use the fact $HZH = X$ and apply our circuit for correcting bit flips with the error zone flanked by $H^{\otimes 3}$. As a bit flip here corresponds to a phase flip in the previous circuit, it passes through as a phase flip did before.

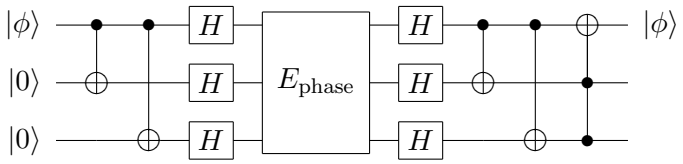

Figure 3: Phase flip correction

We can handle combined errors by concatenating both codes as in Figure 4. We use the bit flip code internally to correct a bit flip and transfer a phase flip onto the first of the three qubits in the block. The dashed line in Figure 4 encircles one of the internal bit flip codes. There are three such blocks, namely one for each of the qubits of the external code, fore which we use our phase flip code.

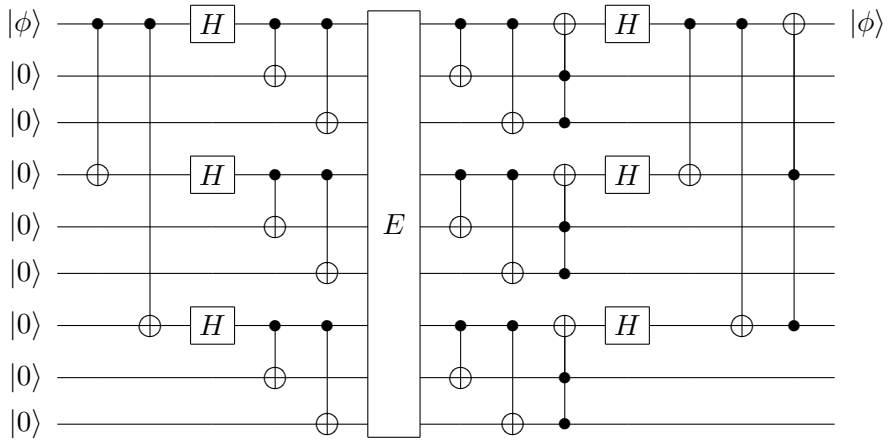

Figure 4: Full Single Qubit Error Correction

The resulting 9-qubit code can correct a single bit flip X, a single phase flip Z, and their combination Y. By linearity, this means that the code can correct *any* single-qubit error E, as any such E can be written as a linear combination of I, X, Y, and Z. Here we can see the magic of quantum linearity at work – although there is a continuum of possible single-qubit errors, is suffices to correct a discrete set (bit and phase flips and their combinations) in order to correct all.

Next lecture we will discuss a different method of single qubit error correction that uses only 7 qubits to represent a single logical qubit.