0.1 Quantum Computing Model

As it relates to this paper, we can think of a quantum computer as a generalization of a classical randomized computer. To begin, let us define a formalism for talking about randomized circuits. We will consider randomized circuits which act on n bits (including additional storage bits in the initial count), and whose state remain in n bits. A fact which will be important to us is that the "not" and "control not" gates (called X, CX respectively) are universal for deterministic classical computation. It follows that, to get universal randomized computation all we will need is a "randomize" gate, which takes a bit and applies X with 50% probability (called an R gate). Now, we conceptualize the state of the computer as a vector $\vec{v}$ of the form

$$\vec{v} \in \mathbb{R}^{2^n} \tag{1}$$

$$\forall v_i, v_i \geq 0 \tag{2}$$

$$\sum_i v_i = 1 \tag{3}$$

Our convention will be that a basis vector $\vec{e}_m$, which has a 1 in the m-th index corresponds to the bit string given by m in binary. A probabilistic mixture of multiple bit strings is represented by a vector with multiple non-zero entries corresponding to probabilities; this is why the values must sum to 1. We now note that the gates we wrote down all correspond to linear transformations on such vectors. We can write the matrices (noting that CX acts on 2 bits, the rest on 1)

$$R = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \tag{4}$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{5}$$

$$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{6}$$

We can construct gates acting on the i-th bit in an n bit state by tensoring the appropriate number of identites, e.g. flipping the i-th bit corresponds to the gate

$$X_i = \left(\bigotimes_{0 \leq j < i} I \right) \otimes X \otimes \left(\bigotimes_{i < j < n} I \right) \tag{7}$$

Now, a circuit C can be thought of as a matrix M_C which is $2^n \times 2^n$; circuits which we can reach as the product of $poly(n)$ many gate-matrices correspond to efficient circuits. We can think of the actual output of C given starting string s, as being drawn from the distribution given $M_C \vec{e}_s$.

We can now interpret quantum computing as a generalization of randomized computing in the following way. We will say that a quantum circuit acts on n qubits (quantum bits) and produces an output state on n qubits. Now, instead of keeping track of the randomized state by a real vector, we will have a complex vector denoted $|\psi\rangle$ (the meaning of $|i\rangle$ is the i-th column basis vector). This will be subject to

$$|\psi\rangle \in \mathbb{C}^{2^n} \tag{8}$$

$$= \sum_{i=0}^{2^n - 1} \alpha_i |i\rangle \sum_i |\alpha_i|^2 = \qquad 1 \tag{9}$$

One can see by standard linear algebra that the set of linear operations preserving these properties for all $|\psi\rangle$ are the unitary matrices of size $\mathbb{C}^{2^n} \times \mathbb{C}^{2^n}$. We will state without proof that the following one or two qubit unitaries can be considered atomic gates

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{10}$$

$$P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\theta} \end{pmatrix} \tag{11}$$

$$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{12}$$

We can "physically" interpret the state as describing a probability distribution on bit string states, with probability of string i given by $|\alpha_i|^2$. However, unlike in probabilistic computing we can now have "destructive interference", wherein multiple computation paths can have some negative and some positive amplitudes such that over-all they cancel out. Usually, the input and output of a quantum program is a classical string, so we feed in a basis vector and measure at the end, obtaining each state with probability given by the rule above.

A "quantum program" is just an ordered list of these operations, along with the qubits each acts on, and efficiently quantum computable functions are ones which have an efficient quantum algorithm (say that gets the right answer with probablity at least $2/3$). Efficient quantum programs are a super-set of efficient classical programs, as they contain in their gate set a CX and X gate (this is non-obvious from our set given; though it is true). Additionally, if we consider applying an H to a pure qubit and then immediately measuring, we obtain a random output. In this way, we can see that efficient quantum programs are also a superset of efficient random programs. It may not be obvious that they are more powerful than random programs, as the only novel ability discussed so far is destructive interference. We will see in later sections how this property can be exploited for computational speed-up. When a vector $|\psi\rangle$ has many non-zero entries, it is referred to as a "coherent superposition", and it is important to understand that this is fundamentally different from a probabilistic mixture. A state like the following

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{13}$$

Is a superposition state. We note that, on measurement, qubit 0 and qubit 1 will both be randomly distributed, but perfectly correlated such that they have the same value. In addition, even if we apply an H gate to each qubit before measuring we see that they are still perfectly correlated; in this sense there is no way to "scramble away" the correlation without measuring. This property is called entanglement, and it is in some sense stronger than classical correlation; we will use it extensively when constructing quantum algorithms.