CS 682: Quantum Computing	Lecture 1: July 29, 2008
Lecturer: Piyush P. Kurur	*Scribe: Lovy Singhal*

1.1 What do we mean by computing?

By the word *computation*, we refer to processing of some input data by a *machine* to produce certain desirable output. A *machine* is any physical device which carries out the *instructions* given to it, employing the existing laws of physics. An *instruction* is a sequence of basic operations, each of which require zero-intelligence and finite time to be carried out. We would like to stress once again that computation cannot rely on supernatural powers to be perfomed or on anything which is beyond the purview of current physics. Conversely, every physical process can be thought of as some kind of computation. The last statement encourages one to think of every physical device as an instance of a computer in addition to suggesting that any physical phenomena can be exploited for computational purposes.
We desire a computational model which should be able to capture all the possible set of operations performed by any sort of a 'real-world' computer. Hence, it certainly cannot be one from amongst them. Therefore, we should turn our attention to things which are in some sense "abstract".

1.2 Turing machine

Turing machines were conceptualised as a 'thought-machine' by Alan Turing in 1936. The basic sketch of a Turing machine consists of the following:

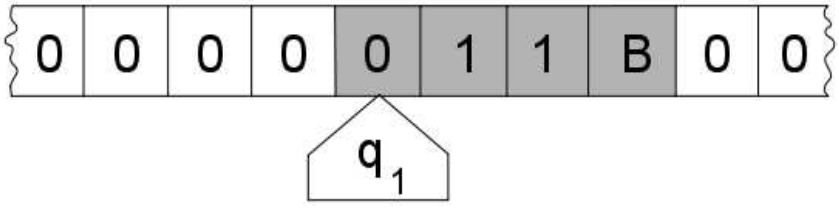

Figure 1.1: A sketch of a Turing machine[1]

1. A **Tape** which is made up of cells, each cell having some 'letter' written on it belonging to some finite alphabet. The tape is infinitely extendible in both directions

2. A **Head** which is capable of reading and writing letters on the tape and also in moving the tape left or right one (and exactly one) cell at a time.

3. A finite **set of instructions** that dictates the action of machine depending on its initial state and the letter on the tape, by telling it whether to move the tape to the left or right and/or to change its own state and/or to overwrite a new letter on the tape.

4. A **register** that stores one of the finitely many possible states that the Turing machine is in.

In all their simplicity, Turing machines are capable of simulating any computing process.

1.3 Dawn of Quantum Era

All the presently available 'computers' are based on the concepts of classical physics. But as it turns out, the world is not completely classical – it is quantum-mechanical at sub-atomic level. Therefore, in our pursuit of the ultimate computing model, the next logical step might be to look if quantum phenomenon can be used for the purpose of computing. Interestingly, one of the pioneers of computing, von Neumann worked on quantum probability but considered quantum mechanics to be a nuisance rather than an aide in computation. The general view that prevailed till not quite long ago was that computing deals with accurate answers to the questions asked, while quantum effects are all about uncertainty and hence can be of no help to compute. But, as it turns out, this is not quite the case.

To erase any possible misunderstanding that might arise, we should make clear that a quantum computer is capable of doing exactly the same operations as any classical computer can, and no more. But, the time complexity of quantum algorithms can be considerably lesser than the corresponding algorithms based on classical computers. As for example, let us take the innocuous looking problem of finding a prime factor p of a composite integer N. Here, our input size is $\log N$ and not N. Hence, ideally we would like to have a polynomial-time algorithm in $\log N$. We shall, in general,

[1]Image Credits: Wikipedia (`http://en.wikipedia.org/wiki/Image:Turing_machine_2b.svg`)

refer to such algorithms which are polynomial-time in input size as *fast algorithms*. In 1994, Peter Shor gave a quantum algorithm for the above-mentioned *factorization problem* which is polynomial time in input size, while the best-known classical factoring algorithm so far is only exponential time in $\log N$.