# Quantum Computing Review

Amir Hossein Jabbari

August 4, 2002

## 1 Bits and Qubits

Consider an electron in a hydrogen atom;

- in a *classical system*, this electron is either in it's normal orbit $1S^1$ (ground = 0), or in the exited position in the next orbit $2S^1$(exited =1).
- in a *quantum mechanical system*, it is not known in which state the electron is. In fact the electron is in a linear superposition of the two states, and is called a qubit[4].

  $$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad \alpha, \beta \in C \text{ and } \alpha^2 + \beta^2 = 1$$

  To know the electron's state, measurement in the $\{|0\rangle, |1\rangle\}$ is needed. Where measurement yields to $|0\rangle$ with the probability $\alpha^2$ , and $|1\rangle$ with the probability $\beta^2$. Moreover, measurement process will alter the state of the qubit. The new state is exactly the outcome of the measurement. That means, no more information about $\alpha$ and $\beta$ could be collected by repeating the measurement[1].

## 2 Quantum Teleportation [5]

Alice wants to send $\chi = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ in a public channel:

1. Alice applies *Controlled-Not*[1][2] operation on her two qubits.

   $$\chi(C-NOT) = (\alpha|0\rangle + \beta|1\rangle)(|00\rangle + |11\rangle) = \alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle$$

2. Alice applies *Hadamard Transformation*[2][2] on first qubit.

   $$\chi(C-NOT)(H) = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|100\rangle + \alpha|011\rangle + \alpha|111\rangle + \beta|010\rangle - \beta|110\rangle + \beta|001\rangle - \beta|101\rangle)$$

[1] Controlled-Not is unitary operation that takes two qubits, and uses the first as control and the second as target. If the first is 0, do nothing, If the first is 1, then NOT the second.

[2] Hadamard Transformation:

$$\begin{array}{l} |0\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{array}$$

3. Alice can rewrite the first two bits in the Bell basis:

$$(|00\rangle + |11\rangle)(\alpha|0\rangle + \beta|1\rangle)+$$
$$(|00\rangle - |11\rangle)(\alpha|0\rangle + \beta|1\rangle)+$$
$$(|01\rangle + |10\rangle)(\alpha|0\rangle - \beta|1\rangle)+$$
$$(|01\rangle - |10\rangle)(-\alpha|0\rangle + \beta|1\rangle)$$

4. Alice sends Bob the information about which Bell pair she has used.

5. Bob performs a correct rotation resulting in $(\alpha|0\rangle + \beta|1\rangle)$.

# 3 Quantum Physics and Church-Turing Machine

## 3.1 Classic Church Turing machine

*Turing Machine* could be imagined as a machine with two parts:

1. A *tape* with infinite length in either side.

2. A *Head* which can move back and forth over the tape. It can also read, write and delete the contents of the tape.

Here is the working instruction of *the machine*:

1. The tape has been partitioned. Each partition can hold either a 1 or a 0.

2. in each step the head could do one of the following:

   (a) halt.

   (b) move one partition to left.

   (c) move one partition to right.

   (d) delete the content of a partition and write over it.

A program is a set of instructions. Whenever no instruction is applied, *the Turing machine* halts.

As an Example, below a *Turing machine* with instruction to compute the *parity function* is presented:

$$f_{(n)} = \begin{cases} 1 & \text{if } n \text{ is odd} \\ 0 & \text{otherwise} \end{cases}$$

The set of instructions for this function is:

| *Curr. State* | *curr.Value* | *next State* | next Value |
|---|---|---|---|
| 1 | 1 | 0 | 2 |
| 2 | 0 | R | 3 |
| 3 | 0 | 1 | 5 |
| 3 | 1 | 0 | 4 |
| 4 | 0 | R | 1 |

The above instruction means:

1. If the program is at state 1 and scanning 1, then write 0 and go to state 2.
2. If the program is at state 2 and scanning 0, then head moves one partition to right and goes to state 3.
3. If the program is at state 3 and scanning 0, then write 1 and goes to state 5.
4. If the program is at state 3 and scanning 1, then write 0 and goes to state 4.
5. If the program is at state 4 and scanning 0, then head moves one partition to right and goes to state 1.

Now lets try $f_{(2)}$ on the *Turing machine*:
$2 = 10_{(binary)}$
Starting with State 1 and reading first bit from left to right:

1. At state 1 and scanning 1, therefore write 0 and go to state 2[3].

   $\bar{0}0$

2. At state 2 and scanning 0, therefore the head moves one partition to right and the program goes to state 3.

   $0\bar{0}$

3. At state 3 and scanning 0, therefore head writes 1 and goes to state 5.

   $0\bar{1}$

4. At state 5 and scanning 1.

The output of the machine is:
$f_{(2)} = 1$

### 3.2 Church-Turing Theses and Quantum Computing

The model described above is a classic Turing-machine, in every state it is known that what the head of the machine will read. *Probabilistic Turing Machine* works the same way, except in every state the head either reads 1 or 0, with some probability[3].

**Theorem 3.1.** *The Modern Church-Turing theorem states that:*

Any reasonable model of computation can be efficiently simulated by a probabilistic Turing machine.

*However, as Feynman pointed out, it appears that the evolution of quantum systems cannot be efficiently simulated with a classical Turing machine,. Not even with a probabilistic one, since computational trajectories in a probabilistic Turing machine cannot interfere with each other(as is possible in a quantum computer).*

[3] The bar over 0 represents the position of the head.

## References

[1] *A. Barenco. Quantum physics and computers, 1996.*

[2] *Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman H. Margolus, Peter W. Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation.* Physical Review A*, 52(5):3457–3467, 1995.*

[3] *David Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer.* Proceedings of the Royal Society of London Ser. A*, A400:97–117, 1985.*

[4] *A. Ekert, P. Hayden, and H. Inamori. Basic concepts in quantum computation.*

[5] *U Vazirani. Lectures in quantum computing, lecture 1.*