Quantum computing

Elisa Pettinà
telegram: @ElisaPettina
Github: QuantumComputing

March 12, 2022

Contents

Chapter 1

Introduction: from classical to quantum

1.1 Classical computing

We'll start our discussion about quantum computing by first reviewing fundamental classical computers' notions.

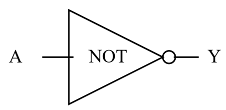
1.1.1 Bits and logical gates

Classical computers are made by

- Elementary units: **bits**, that can take values $0, 1$;
- Ellementary operations carried out by **logical gates**.

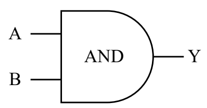
Examples of the usage of these two elements are the

- The NOT gate



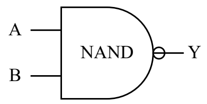
Truth table	
A	Y
0	1
1	0

- The AND gate



Truth table		
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

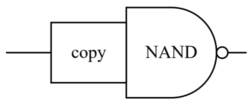
- The NAND gate



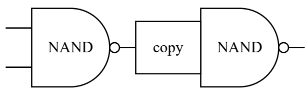
Truth table		
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

1.1.1.1 Universality and Turing machine

The concept of universality yields a set of elementary objects that can do all sort of comptutation inside a classical computer. The NAND gate with the copy procedue are togheter universal gates, since all gates can be cosntructed from these two.
In the pictures below the construction of the NOT and AND gate are depicted, using ongli the copy and the NAND.

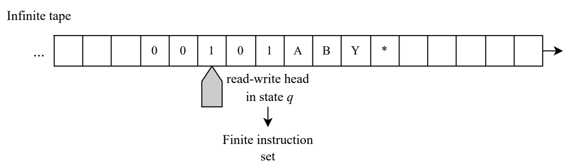


Input	Copy	NAND
0	00	1
1	11	0



Input	NAND	Copy	NAND
00	1	11	0
01	1	11	0
10	1	11	0
11	0	00	1

Since the NAND and copy operations can be used to describe all sort of (classical) computations, they also provide for the costruction of the Turing machine.
A Turing machine is a mathematical model of computation that defines an abstract machine that manipulates symbols on a strip of tape according to a table of rules.Despite the model's simplicity, given any computer algorithm, a Turing machine capable of implementing that algorithm's logic can be constructed[1].



The machine operates on an infinite memory tape divided into discrete "cells". The machine positions its "head" over a cell and reads the symbol there. The read/write head is now in state q. Given the state q and the symbol, the machine proceeds with the given finite instruction set:

[1] wikipedia/Turing_Machine

modifies the symbol, the head moves and q is adapted. Based on the observed symbol and the machine's own state, either proceeds to another instruction or halts computation.
However, the very same definition of the turing machine proves the existance of its fundamental limits. For example, it cannot solve for problems that are not deducidable. The *halting problem* is a well known case: it states that it does not exist an algorithm f that cand etermine for any other agorithm g if g will eventually terminate or run forever. Algorithm shows as an absurd an algorithm f that checks wether g terminates:

: f(algorithm g)

```
if f(g) = terminate then
    while true do
        end
else
    return
```