QuantumComputation

Andrew Tulloch

Contents

CHAPTER 1

Significance of Quantum Computation

(i) Fundamentally:
  (i) Can view physics as information processing — physical evolution is equivalent to updating parameters describing states - thus computation.
  (ii) Can view computation as physics - classical bit 0/1 are two distinguishable states of a physical system.
(ii) Technological
  (i) Moore's law
  (ii) Nano-technology - coherent controlled manipulation of quantum systems
  (iii) Further issues - information security (quantum cryptography)
(iii) Theoretical
  (i) New modes of computation, allowed by quantum vs classical effects - implications for computational complexity.

1. Computability vs non-computability

(i) Given N (integer), is it prime? **computable**.
(ii) Given a polynomial with integer coefficients - e.g. $2x^2y - 17zw^{19} + x^3$, does it have a root in the integers? **non-computable**.

Quantum computing - all laws of quantum mechanical evolution are computable on a classical computer, so QC cannot compute any classically non-computable problem. But computability is not equivalent to computational complexity.

2. Computational Tasks

Given an input bit string $x = i_1 \dots i_n \in B_n$, $B = \cup_{n=1}^{\infty} B_n$ (all finite length bit strings). A language $L \subseteq B$.

Definition 1.1 (Decision problem). Given $x \in B$ is $x \in L$? Output is 1 bit 0/1, yes/no, accept/reject.

Question 1.2. *How hard is it to solve the problem as a function of n, the size of the input in bits?*

A computational model is a classic circuit model - for each n, we have a circuit C_n of AND/OR/NOT gates. The computational steps are this gate. Note this is a universal set - can make any Boolean function $f : B_n \to B_n$ as a function of AND/OR/NOT gates. Full computation is a circuit family $(C_1, C_2, \dots)$.

In a random model of classical computation, we allow further random bits of input into the gate.

Complexity/hardness of the computation is measured by the consumption of resources - time (number of computational steps as a function of n), and space (number of bits needed/work space needed).

Polynomial time complexity classes. Let $T(n)$ be the maximum number of steps for any input of size n - (in a circuit model, equivalent to the number of gates, and so circuit size). The main question is - does $T(n)$ grow polynomially

....

Following along from the notes

....

3. Simon's Algorithm

...

4. Hidden Subgroup Problems

Known:

(i) HSP can be solved for any abelian group.
(ii) Not known for general non-Abelian group, but
    (i) $O(\log |G|)$ random $|gH\rangle$'s suffice to determine H. Not known how to extract into H...
    (ii) Normal subgroup $H <| G$, then can solve HSP.

Other problems reduce to HSP.

(i) Discrete logarithm is abelian HSP.
(ii) Graph isomorphism is Hiddn Subgroup problem for non-Abelain G and non-normal H, so don't know how to solve.

Bibliography