Quantum Computing

Vaughn Sohn

November 15, 2024

Contents

Chapter 1

Quantum Algorithms

Lecture 9

1.1 Introduction

7 Oct. 10:30

이번 챕터에서 우리는 *Quantum Algorithm*에 대해 다루고자한다. Quantum algorithm은 quantum circuit이나 quantum computer에서 구현되는 알고리즘을 지칭한다. Classical computer가 어려운 문제를 해결하기 위하여 만들어진 것처럼, quantum algorithm에 대해서 공부하고 새로운 방식을 고안하는 것은 quantum computer의 동작방식과 quantum computer의 한계를 분석하기 위한 중요한 과제이다. Quantum computer라는 개념이 등장하고 나서부터 지금까지 많은 종류의 quantum algorithm들이 고안되어 왔다. 이번 강의에서 다루고자하는 quantum algorithm은 다음과 같다.

- Elementary quantum algorithms
- Hamiltonian simulations
- Quantum Fourier transform
- Phase estimation
- Quantum search algorithm (Grover search algorithm)
- Amplitude amplification / estimation algorithms
- HHL algorithm

1.2 Elementary quantum algorithms using quantum parallelism

1.2.1 Deutsch's algorithm

Lecture 10

1.2.2 Deutsch-Jozsa algorithm

14 Oct. 10:30

1.3 Hamiltonian simulations

Lecture 11

16 Oct. 10:30

1.4 Quantum Fourier transform

1.5 Phase estimation

1.5.1 Phase estimation

Lecture 12

1.5.2 Performance

28 Oct. 10:30

Lecture 13

1.6 Applications of phase estimation

30 Oct. 10:30

1.6.1 Order-finding algorithm

Order-finding

Uncomputation

The continued fraction expansion

Performance

1.6.2 Shor's algorithm: factoring

Lecture 14

1.7 Applications of the QFT

4 Nov. 10:30

1.7.1 Period-finding

1.7.2 Discrete logarithm

1.7.3 Hidden subgroup problem

1.8 Quantum search algorithms

1.8.1 Grover operator

1.8.2 Grover search algorithm

1.8.3 Performance

Lecture 15

1.8.4 Example: Classical circuit-SAT problem

6 Nov. 10:30

1.8.5 Amplitude amplification

1.9 Amplitude estimation algorithm (=Quantum counting)

1.10 HHL (Harrow–Hassidim–Lloyd) algorithm

Lecture 16

1.11 Optimality of the quantum search algorithm

8 Nov. 17:00

Chapter 2

Introduction to Computational Complexity

Lecture 16

2.1 Introduction

8 Nov. 17:00

Lecture 17

2.2 The class NP: Reducibility and completeness

11 Nov. 17:00

2.2.1 P and NP problems

2.2.2 Reducibility and NP-completeness

2.2.3 Boolean formula and Cook-Levin theorem

2.3 Quantum complexity

2.3.1 Probabilistic algorithms

2.3.2 Quantum algorithms

2.3.3 BQP vs PSPACE

Appendix

Appendix A

Useful Environments for the Note

A.1 Useful Environment

We now see some common environment you'll need to complete your note.

Definition A.1.1 (Natural number)**.** We denote the set of *natural numbers* as $\mathbb{N}$.

Lemma A.1.1 (Useful lemma)**.** Given the axioms of natural numbers $\mathbb{N}$, we have

$$0 \neq 1.$$

An obvious proof. Obvious. ■

Proposition A.1.1 (Useful proposition)**.** From Lemma A.1.1, we have

$$0 < 1.$$

Exercise. Prove that $1 < 2$.

Answer. We note the following.

> **Note.** We have Proposition A.1.1! We can use it iteratively!
>
> With the help of Lemma A.1.1, this holds trivially. ⊛

Example. We now can have $a < b$ for $a < b$!

Proof. Iteratively apply the exercise we did above. ⊛

Remark. We see that Proposition A.1.1 is really powerful. We now give an immediate application of it.

Theorem A.1.1 (Mass-energy equivalence)**.** Given Proposition A.1.1, we then have

$$E = mc^2.$$

Proof. The blank left for me is too small,[a] hence we put the proof in appendix. ■

[a] https://en.wikipedia.org/wiki/Richard_Feynman

From Theorem A.1.1, we then have the following.

Corollary A.1.1 (Riemann hypothesis)**.** The real part of every nontrivial zero of the Riemann zeta function is $\frac{1}{2}$, where the Riemann zeta function is just

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} = \frac{1}{1^s} + \frac{1}{2^s} + \frac{1}{3^s} + \cdots .$$

Proof. The proof should be trivial, we left it to you. ■

TODO mark

As previously seen. We see that Lemma A.1.1 is really helpful in the proof!

Internal Link

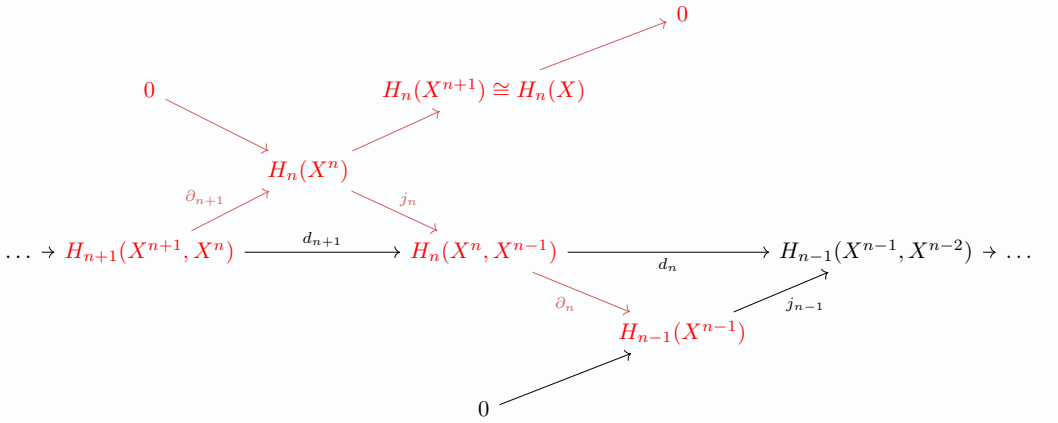
You should see all the common usages of internal links. Additionally, we can use citations as citation [NC01], which just link to the reference page!

A.2 Commutative Diagram

We can use the package `tikz-cd` to draw some commutative diagram.

Example. The cellular homology agrees with singular homology.

Proof. The following commutative diagram shows everything.



⊛

A.3 Fancy Stuffs

With this header, you can achieve some cool things. For example, we can have multiple definitions under a parent environment, while maintains the numbering of definition. This is achieved by `definition*` environment with `definition` inside. For example, we can have the following.

Definition. We have the following number system.

Definition A.3.1 (Rational number)**.** The set of *rational number*, denote as $\mathbb{Q}$.

Definition A.3.2 (Real number)**.** The set of *real number*, denote as $\mathbb{R}$.

Definition A.3.3 (Complex number)**.** The set of *complex number*, denote as $\mathbb{C}$.

Note. And indeed, we can still reference them correctly. For instance, we can use rational numbers to define real numbers and then further use it to define complex numbers.

Furthermore, we can completely control the name of our environments. We already saw we can name definition, lemma, proposition, corollary and theorem environment. In fact, we can also name remark, note, example and proof as follows.

Example (Interesting Example). We note that $1 \neq 2$!

Note (Important note). As a consequence, $2 \neq 3$ also.

Remark (Easy observation). We see that from here, we easily have the following theorem.

Theorem A.3.1 (Lebesgue Differentiation Theorem). Let $f \in L^1$, then

$$\lim_{r \to 0} \frac{1}{m(B(x,r))} \int_{B(x,r)} |f(y) - f(x)| \, \mathrm{d}y = 0$$

for a.e. x.

An obvious proof of Theorem A.3.1. Obvious. ■

As we can see, specifically for the `proof` environment, we allow `autoref` and `hyperref`. One can actually allow all example, note and remark environment's name to use reference, but I think that is overkilled. But this can be achieved by modify the header in an obvious way.[1]

[1] This time I mean it!

Bibliography

[NC01] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Vol. 2. Cambridge university press Cambridge, 2001.