Seminar 2

Formalisms: QM & Computing

Some Important Announcement

- I have to quit SNU as I have to undergo my visa application.
- Still, I will have my student ID card so hopefully I can request admittance claiming my card doesn’t work…?
- (Seriously, this place is so good!)

Schedule

- 1. Introduction to Quantum Computing
- 2. Formalisms in QM & Computing - Today
- 3. Quantum Computations
- 4. Realisations of Quantum Computers
- 5. Quantum Noise, Operations, and Distances
- 6. Error Corrections
- 7. Quantum Entropy & Information Theory

Before going in…

- No Rocket Science today!
- Today, we have three things to do.
- 1. Tensor Products,
- 2. Basics in QM: Projections/POVM/Mixed Ensembles
- 3. Basics in CompSci: Turing Machines/Circuits/Complexity
- Some prerequisites: Basic Linear Algebra/Bra-Ket Formalisms/Graph Theory
- Today’s session shall be short; I expect around 45 min.

Quick Overview on Linear Algebra

- Vector spaces: Usually, we call the columns ‘vectors’.
- Srsly, everyone will know how to add/subtract/multiply matrices.

- Linear Independence? Hermitians? Operators?

Tensor Products

- This is a way of putting vector spaces together to form larger vector spaces.
- $V \otimes W$
- Notations: $|vw\rangle = |v\rangle \otimes |w\rangle$

Properties of Tensor Products

- $z(|v\rangle \otimes |w\rangle) = (z|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z|w\rangle)$
- $(|v1\rangle + |v2\rangle) \otimes |w\rangle = |v1\rangle \otimes |w\rangle + |v2\rangle \otimes |w\rangle$.
- $|v\rangle \otimes (|w1\rangle + |w2\rangle) = |v\rangle \otimes |w1\rangle + |v\rangle \otimes |w2\rangle$.

- Easy?

Linear Operators Acting on V ⊗ W

- Suppose |v⟩ and |w⟩ are vectors in V and W , and A and B are linear operators on V and W.
- $(A \otimes B)(|v\rangle \otimes |w\rangle) \equiv A|v\rangle \otimes B|w\rangle$.
- To ensure linearity, $(A \otimes B)(\sum_i a_i |v_i\rangle \otimes |w_i\rangle) = \sum_i a_i A|v_i\rangle \otimes B|w_i\rangle$

Linear Operators Acting on V ⊗ W

- Indeed, an arbitrary linear operator C mapping V ⊗ W to V ′ ⊗ W ′ can be represented as a linear combination of tensor products of operators mapping V to V ′ and W to W
- $C = \sum_i c_i A_i \otimes B_i$

- Inner Products: Define the inner product as $< \sum_i a_i |v_i\rangle \otimes |w_i\rangle \,, \sum b_j |v_j{}'\rangle \otimes |w_j{}'\rangle > = \sum_{ij} a_i^* b_j \langle v_i | v_j{}'\rangle \langle w_i | w_j{}'\rangle$

Explicit Calculation of the Tensor Product

- Suppose A is an m by n matrix, and B is a p by q matrix.
- The Kronecker Product

$$A \otimes B \equiv \overbrace{\begin{bmatrix} A_{11}B & A_{12}B & \dots & A_{1n}B \\ A_{21}B & A_{22}B & \dots & A_{2n}B \\ \vdots & \vdots & \vdots & \vdots \\ A_{m1}B & A_{m2}B & \dots & A_{mn}B \end{bmatrix}}^{nq} \Bigg\} mp .$$

- Example 1. Calculate the Tensor Product of $\frac{1}{2}$ and $\frac{2}{3}$.
- Example 2. Calculate the Tensor Product of the X and Y Pauli matrices.

The Polar and Singular Value Decompositions

- (Polar) Let A be a linear operator on a vector space V . Then there exists unitary U and positive operators J and K such that
- $A = UJ = KU$ where the unique positive operators J and K satisfying these equations are defined by $J \equiv \sqrt{A^\dagger A}$ and $K \equiv \sqrt{AA^\dagger}$. Moreover, if A is invertible then U is unique.
- (Singular Value) Let A be a square matrix. Then there exist unitary matrices U and V , and a diagonal matrix D with non-negative entries such that
- $A = UDV$.

Projective Measurements

- A projective measurement is described by an observable, M
- Of course, M has a spectral decomposition, noted as $M = \sum_m mP_m$
- This concept is simple, if not trivial.
- Let $p(m) = \langle\psi|P_m|\psi\rangle$, then after the measurement gives m, the wave function collapses to $\frac{P_m|\psi\rangle}{\sqrt{p(m)}}$

POVM measurements

- Abbreviation for Positive Operator-Valued Measure
- Measurement Operator M_m: the probability of outcome m is given as $p(m) = \langle\psi|P_m^\dagger P_m|\psi\rangle$
- Define $E_m = P_m^\dagger P_m$, then $\sum_m E_m = 1$ & $p(m) = \langle\psi|E_m|\psi\rangle$
- The set of E_m is known as the POVM.

Incoherent mixtures

- Consider the Stern-Gerlach experiment
- For the + and – states, we DO NOT have information on the phase differences. (This will cause huge problems.)
- What does $|a\rangle = c_{+}|+\rangle + c_{+}|-\rangle$ imply? Can this state explain the terrible situation above?
- This is more with probability and statistics, not wave functions.

Ensemble Averages

- Let's introduce 'probabilities': w_+, w_- in SG experiment.
- Pure Ensemble: Every member can be characterised by the same ket.
- Mixed Ensemble: Roughly speaking, only a fraction of members are represented by the same ket. Let's write the fractions as w_i.
- $\sum_i w_i = 1$.
- Obviously, the states DO NOT have to be orthogonal nor coincide with the dimension of the ket space.
- Example: In spin ½ systems, 50% in z+, 20% in x+, 30% in y+

Density Operator

- Ensemble Average: $[A] = \sum_i w_i \langle \alpha^{(i)} | A | \alpha^{(i)} \rangle = \sum_i \sum_{a'} w_i \left| \langle a' | \alpha^{(i)} \rangle \right|^2 a'$, where a' is the eigenket of A.
- We can look this in other generalised basis kets (suppose b', b'')
- The basic property of the ensemble that DOES NOT depend on the observable can be factored out.
- $[A] = \sum_{b'} \sum_{b''} (\sum_i w_i \langle b'' | \alpha^{(i)} \rangle \langle \alpha^{(i)} | b' \rangle) \langle b' | A | b'' \rangle$
- Density operator: $\rho = \sum_i w_i | \alpha^{(i)} \rangle \langle \alpha^{(i)} |$

Properties of Density Operators

- $[A] = \sum_{b'} \sum_{b''} \langle b''|\rho|b'\rangle \, \langle b'|A|b''\rangle = tr(\rho A)$
- $tr(\rho) = \sum_i \sum_{b'} w_i \langle b'|\alpha^{(i)}\rangle\langle\alpha^{(i)}|b'\rangle = 1$
- $\rho^2 = \rho \; \longrightarrow \mathrm{tr}(\rho^2) = 1$
- We can obviously put the density operator in a matrix form..

Example

- Find the Density Operator in Matrix Form in the following states.
- 1. A completely polarised beam for z+-spin & y+-spin
- 2. Incoherent mixture of 50% z+ and 50% z-. Calculate the ensemble average for S. (Not z-direction!)
- 3. 75-25 mixture of Sz+ and Sx+. Calculate the ensemble averages for Sx, Sy, and Sz.

Time Evolution of Density Operators

- $\rho(t_0) = \sum_i w_i \left|\alpha^{(i)}\right\rangle \left\langle\alpha^{(i)}\right|$
- Let time evolution to happen, and consider that the kets obey the Schrodinger equation!
- $i\hbar \frac{\partial\rho}{\partial t} = \sum_i w_i (H\left|\alpha^{(i)}, t_0; t\right\rangle \left\langle\alpha^{(i)}, t_0; t\right| - \left|\alpha^{(i)}, t_0; t\right\rangle\left\langle\alpha^{(i)}, t_0; t\right|H)$
  $= -[\rho, H]$
- Note that the Schrodinger equation displays OPPOSITE signs when applied to conjugates. Also, this looks quite similar to the Heisenberg equation of motion.
- Classical Analogue (Liouville's Thm): $\frac{\partial\rho}{\partial t} = -[\rho, H]$

Continuum Generalisation

- Change the sigmas to integrals.
- I'm not going to write this (My hands hurt.).
- $\langle x''|\rho|x'\rangle = \sum_i w_i \psi_i(x'')\psi_i^*(x')$
- Trivial?

Quantum Statistical Mechanics: Entropy

- Define $\sigma = -\mathrm{tr}(\rho ln\rho)$
- Okay, let's only consider the diagonal cases. (If not, it shall get messy…)
- Then, $\sigma = -\sum_k \rho_{kk} ln\rho_{kk}$
- Example. Calculate σ for completely random ensemble and pure ensemble. (N states)
- Wait, can't we define $S = k\sigma$? (k is Boltzmann const, but actually it can be any constant!)

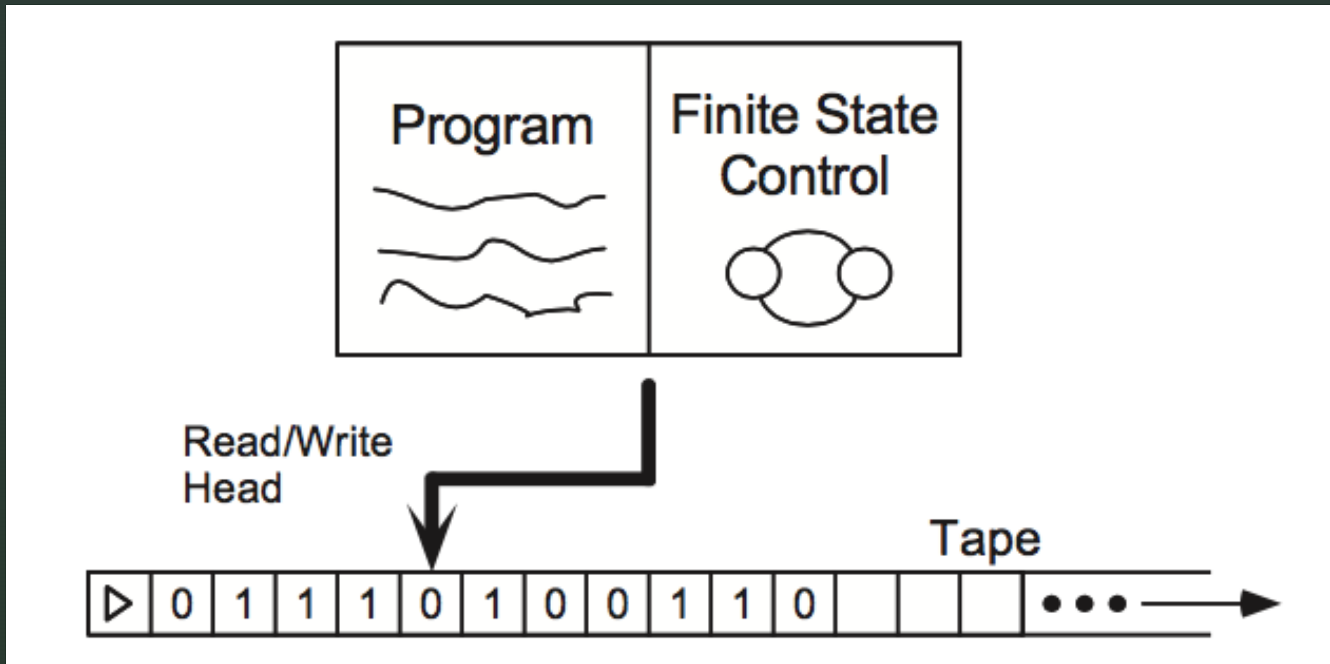
Four Parts of the Turing Machine

- (a) a program, rather like an ordinary computer;
- (b) a finite state control, which acts like a stripped-down microprocessor, co-ordinating the other operations of the machine;
- (c) a tape, which acts like a computer memory;
- and (d) a read- write tape-head, which points to the position on the tape which is currently readable or writable.

Four Parts of the Turing Machine

- Finite State Control
- Consists of a finite set of internal states $q_1, q_2 \ldots q_m$
- m is a variable; sufficiently large m does NOT alter the abilities of the machine for this effect.
- It provides temporary storage off-tape, and a central place where all processing for the machine may be done.
- q_s & q_h : Denotes start and end of the execution

Four Parts of the Turing Machine



Programming in the Turing Machine

- finite ordered list of program lines of the form ⟨q, x, q′, x′, s⟩
- q, q’ are the states; x, x’ are the alphabets. S denotes the next action.
- 1. Find the state which internal state is q and the alphabet is x.
- 2. If you can’t, the state goes q_h and terminated. Else, change the internal state to q’ with alphabet x’.
- 3. Proceed as s dictates.

Example

- What will this programme compute?
- Ans: Constant function f(x)=1
- Will anyone try?

$$
\begin{aligned}
1 &: \langle q_s, \triangleright, q_1, \triangleright, +1 \rangle \\
2 &: \langle q_1, 0, q_1, b, +1 \rangle \\
3 &: \langle q_1, 1, q_1, b, +1 \rangle \\
4 &: \langle q_1, b, q_2, b, -1 \rangle \\
5 &: \langle q_2, b, q_2, b, -1 \rangle \\
6 &: \langle q_2, \triangleright, q_3, \triangleright, +1 \rangle \\
7 &: \langle q_3, b, q_h, 1, 0 \rangle.
\end{aligned}
$$

Church-Turing Thesis

- The class of functions computable by a Turing machine corresponds exactly to the class of functions which we would naturally regard as being computable by an algorithm.
- No exceptions found to date.

- Quiz. Can anyone construct the Turing machine with TWO tapes?

Universal Turing Machines

- Thm. Two-tape Turing machines can simulate One-tape Turing machines.
- Generalisation: There is a universal Turing machine that can simulate an arbitrary Turing machine.
- I will not go on with the construction. (Little out of scope..)

The Entscheidungsproblem

- Is there an algorithm to decide all the problems of mathematics?
- Ans: No.
- Counterexample: The Halting Problem
- Explanation: does the machine with Turing number x halt upon input of the number y?

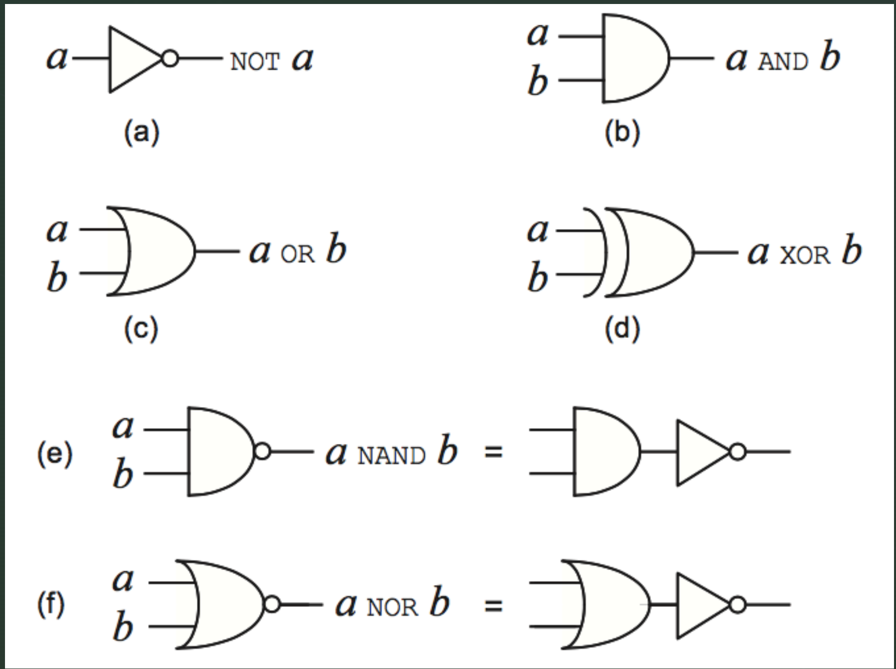
The Halting Problem

```
TURING(x)

y = HALT(x)
if y = 0 then
        halt
else
        loop forever
end if
```

- Define h(x): 1 if halts, 0 if not halts if the input is x
- If there is an algorithm to solve the halting problem, then there surely is an algorithm to evaluate h(x) (Call it HALT(x))
- Since HALT is a valid program, TURING must also be a valid program, with some Turing number *t*.
- By def, h(t)=1 if and only if TURING halts at t.
- Programme: halts when h(t)=0

Circuits



Additional Gates in Classical Computations

- FANIN & FANOUT: I explained those last class.
- CROSSOVER: The value of two bits are interchanged.

- Not a gate, but the preparation of extra ancilla or work bits, or to allow extra working space during the computation is allowed.

Half & Full Adders

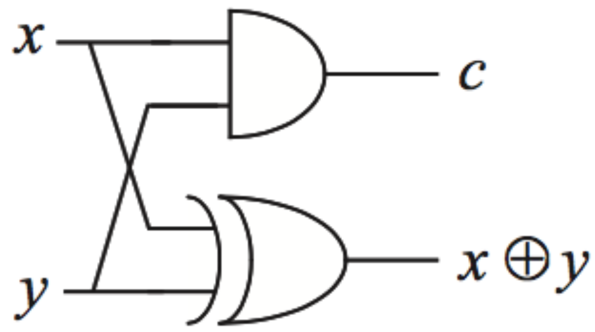


Figure 3.5. Half-adder circuit. The carry bit c is set to 1 when x and y are both 1, otherwise it is 0.

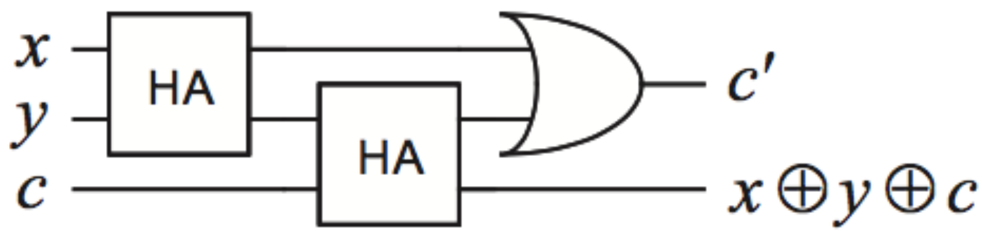


Figure 3.6. Full-adder circuit.

Universality of NAND

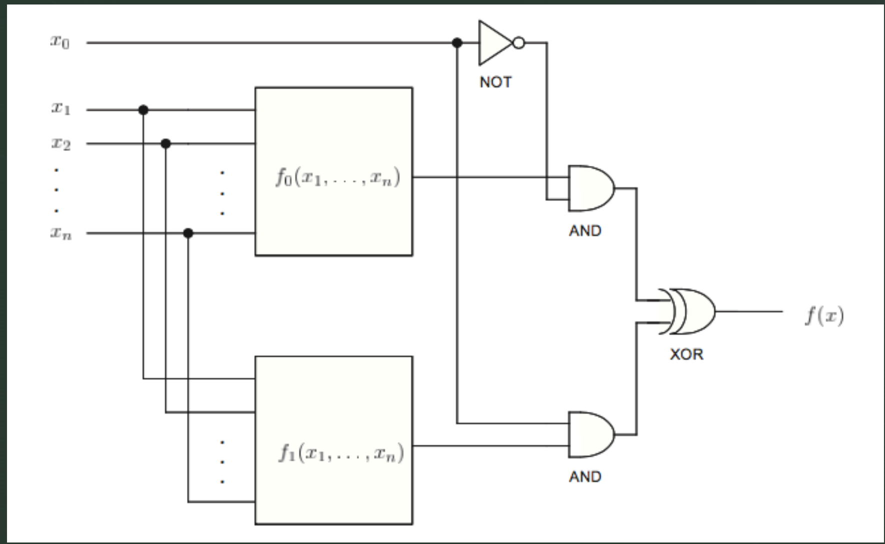
- Step 1. Boolean Functions.
- Boolean has 4 operations: Identity, Bit Flip (NOT), changing input to 0 (AND), changing input to 1 (OR)
- Let’s use induction from this!

Universality of NAND

- Suppose that any function on n bits may be computed by a circuit, and let f be a function on n + 1 bits.
- Define f0 and f1: $f_0 = f(0, x_1, x_2 \ldots x_n), f_1 = f(1, x_1, x_2 \ldots x_n)$
- These are n-bits, so they are computable with circuits.

Universality of NAND

- Depending on the first input, we can make the output the correct answer:



Universality of NAND

- Step 3.
- Universality requires: Wires, Ancilla Bits, FANOUT, CROSSOVER, AND, XOR, and NOT gates.
- These all can be simulated with NAND. (I won't do it.)
- Therefore, it is proved.

Some Extras: Quantum Complexity Theory

- How much resources (time included!) do we need to do a certain task? How efficiently can we do it?
- Strong Church-Turing Thesis: Any model of computation can be simulated on a probabilistic Turing machine with at most a polynomial increase in the number of elementary operations required.
- Shannon (1949): For any $n \geq 2$, there is an n-ary boolean function f such that no boolean circuits with $2^n/(2n)$ or fewer gates can compute it.

The P-Class

- P: class of computational problems that can be solved quickly on a classical computer
- *Quickly ~ In polynomial time
- Most tasks we know how to do will probably be P.

NP-Class

- (1) If $x \in L$ then there exists a witness string w such that M halts in the state qY after a time polynomial in |x| when the machine is started in the state x-blank-w.
- (2) If $x \notin L$ then for all strings w which attempt to play the role of a witness, the machine halts in state qN after a time polynomial in |x| when M is started in the state x-blank-w.
- Factoring: Given a composite integer m and l < m, does m have a non-trivial factor less than l?

Completeness

- There is a language L in the complexity class which is the 'most difficult' to decide, in the sense that every other language in the complexity class can be reduced to L.
- Not all complexity classes have complete problems.
- P-complete definitely exists.
- Example for NP-complete: Circuit Satisfiability Problem (CSAT, Cook-Levin Theorem)

PSPACE-Class

- PSPACE: problems which can be solved using resources which are few in spatial size, but not necessarily in time
- It is easy to see that P and NP are in PSPACE; we don't know whether non P-complete problems are in PSPACE
- Thm. the class of problems solvable on a quantum computer in polynomial time is a subset of PSPACE. (Not now!)
- So, if P=PSPACE, we are doomed.

Other Complexity Classes

- BPP: class of problems that can be solved using randomized algorithms in polynomial time, if a bounded probability of error is allowed in the solution to the problem.
- L: Solvable in Logarithmic Time
- EXP: Solvable in Exponential Time
- MAXSNP: Set of problems possible to efficiently verify approximate solutions to the problem.
- Quiz. Determine the relations between EXP, L, P, PSPACE, and NP.

Landauer's Principle

- Complexity does not necessarily mean time and space; energy is included.
- Landauer's Principle: When a computer erases a single bit of information, the amount of energy dissipated into the environment is at least $k_b T \ln 2$.
- 'Erasing' data (i.e. irreversible) requires energy!

Acknowledgements

- The overall text was amended using Nielsen & Chuang, Quantum Computations and Quantum Information, 10th anniversary edition, Cambridge University Press, 2010.

- The figures used in this presentation is also an excerpt from Nielsen and Chuang.

- This presentation is NOT intended for commercial uses, but for education.