Lecture 1

Course Background

QC ⇌ ML

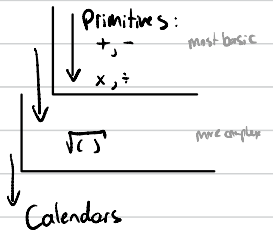
Online Synchronous

- Exercises - completeness only
- Assignments - mostly correctness
- Final Proj - midterm checkin, Final report, Demo day
  Test - Lots of overlap w/ final project

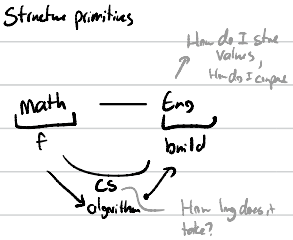
Focus is on application
↳ Don't need to fully understand quantum physics
↳ no quantum 'fluff'

Computation ~ aligned ~ function $f: in \longmapsto out$

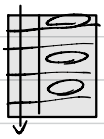
we build computers to implement func.



- Structure primitives



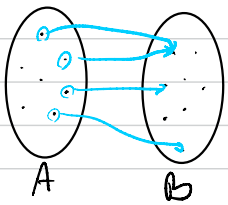
eg. search Hana list
need: ==, >, <



Mapping & Digitizing

$f: A \longmapsto B$

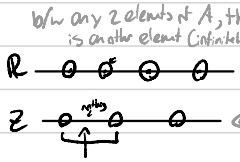
mapping from set A → set B



only one arrow emitted per element in A

$A \subseteq \mathbb{R}^N$ continuous
$A \subseteq \mathbb{Z}$ discrete

b/w any 2 elements of A, there is another element (infinitely)



Can approximate real sets → discrete sets
↳ take small enough steps to discretize
eg. can digitize an image

f:

Eg:

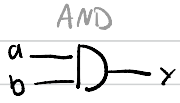
↓ Decomposed into primitive operations

f

set of discrete primitive operations is as simple as computers

Only need 3 functions for a digital computer

AND



$a \in \{0,1\}$
$b \in \{0,1\}$

a	b	Y
0	0	0
0	1	0
1	0	0
1	1	1

Can show any discrete function into a table

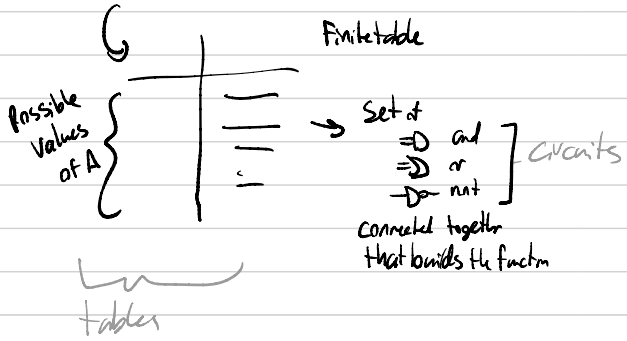
If $a, b \in \mathbb{R}$ or $a, b \in [0,1]$ need uncountably infinite sized table

Dis. Comp

Finite discrete sets and functions using these

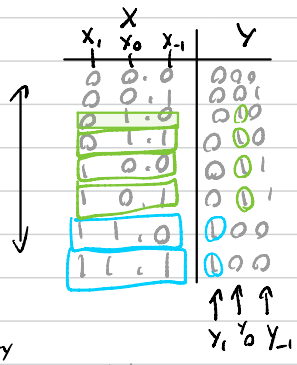
→ <u>computable function</u>

f: discrete A → discrete B
finite finite



e.g. $\sqrt{x}$ $\quad 0 \le x \le 3.5 \pm \frac{1}{2}$

X		Y	
0	0.0	0	0.0
0.5	0.1	0.5	0.1
1	1.0	1	1.0
1.5	1.1	1	1.0
2	10.0	1.5	1.1
2.5	10.1	1.5	1.1
3	11.0	2	10.0
3.5	11.1	2	10.0

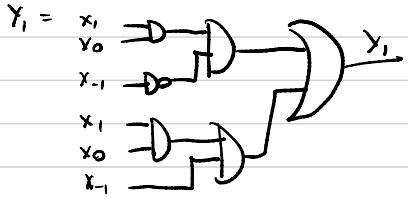


0, 1

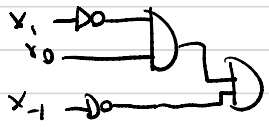
3 binary digits

2^2 2^1 2^0 . 2^{-1}

Y_1 = Y_1 = 1 when

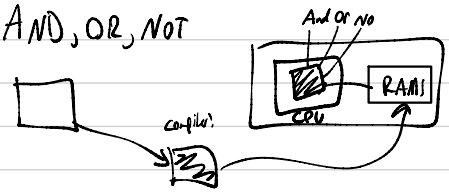


Y_0



Boole, Shannon responsible for this idea

AND, OR, NOT



Stored program computer ~ classic computers

In quantum world we have no CPU, no RAM, only equivalent of AND, OR, NOT

Classical computation:

Any computable function

⇕

(complex) circuit of (multiple) AND, OR, NOT Gates

Gate model of classical computation

Quantum Computers

What are our primitive operations?

- Q. Algos
  - Structure
  - Quantum Circuits

1st Q. Alg is how to map a classical computer alg. to a quantum

Classical Comput.

Primitive operations ⇒ AND, OR, NOT

Comp. Algos ~ Boolean Circuit

- directly → Chip business
- indirectly → compile → intel inside

every algo

Digital comp. solution ≡ Boolean circuit

Classical computers

$a, b, y \in \{0, 1\}$

$a_3 a_2 a_1 a_0 \in \{0000, 0001, \ldots, 1111\}$

AND

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

OR

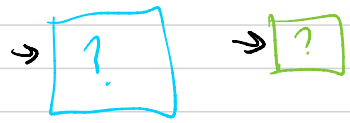
a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

NOT a → b

a	b
0	1
1	0

Structures ~ Circuits based on these ≡ Computer algorithms

Quantum computers



Q Structures ~ Q. Circuits based on these ≡ Q. algorithms