IV. Quantum Computation

IV.1 The circuit model

a) Classical computation

Use of class. computers:

Solve problems $\equiv$ compute functions

$$f: \{0,1\}^n \longrightarrow \{0,1\}^m$$

$$\underline{x} = (x_1, \dots, x_n) \longmapsto f(x_1, \dots, x_n) = \underline{y} = (y_1, \dots, y_m)$$

f depends on problem, $\underline{x}$ encodes instance of problem.

E.g.: Multiplication $(a, b) \longmapsto a \cdot b$

$$\underline{x} = \underbrace{(\underline{x}_1, \underline{x}_2)}_{\text{enc. in binary}} \Longrightarrow f(\underline{x}) = \underbrace{\underline{x}_1 \cdot \underline{x}_2}_{\text{enc. in binary}}$$

Factorization: $\underline{x}$: integer; $f(\underline{x})$: list of prime factors (w/ suitable encoding)

Each problem is encoded by a family of functions

$$f \equiv f^{(n)} : \{0,1\}^n \longrightarrow \{0,1\}^m \; ; \; m = \text{poly}(n); \; n \in \mathbb{N}.$$

What ingredients do we need to compute a general function f?

(i) $f : \{0,1\}^n \longrightarrow \{0,1\}^m$

$$f(\underline{x}) = (f_1(\underline{x}), f_2(\underline{x}), \ldots, f_m(\underline{x}))$$

$$f_k : \{0,1\}^n \longrightarrow \{0,1\}$$

$\Rightarrow$ can focus on Boolean functions $f : \{0,1\}^n \longrightarrow \{0,1\}$.

(ii) Let $L = \{ \underline{y} \mid f(\underline{y}) = 1 \} = \{ \underline{y}^1, \underline{y}^2, \ldots, \underline{y}^\ell \}$

Define $g_{\underline{y}}(\underline{x}) = \begin{cases} 0 \; ; \; \underline{x} \neq \underline{y} \\ 1 \; ; \; \underline{x} = \underline{y} \end{cases}$

Then, $f(\underline{x}) = g_{\underline{y}^1}(\underline{x}) \vee g_{\underline{y}^2}(\underline{x}) \vee \ldots \vee g_{\underline{y}^\ell}(\underline{x})$

"$\vee$" $\equiv$ "logical or":

$$0 \vee 0 = 0$$
$$0 \vee 1 = 1$$
$$1 \vee 0 = 1$$
$$1 \vee 1 = 1$$

associative: $a \vee b \vee c := (a \vee b) \vee c = a \vee (b \vee c)$

(iii) $g_{\underline{y}}(\underline{x}) = \underbrace{(y_1 = x_1)}_{\substack{1 \text{ if } y_1 = x_1 \\ 0 \text{ if } y_1 \neq x_1}} \wedge (y_2 = x_2) \wedge \ldots$

"$\wedge$" ≡ "logical and": $1 \wedge 1 = 1$, else 0.

(iv)
$$(y_i = x_i) = \begin{cases} x_i, & \text{if } y_i = 1 \\ \neg x_i, & \text{if } y_1 = 0 \end{cases}$$

"$\neg$" ≡ "logical not": $\neg 1 = 0$, $\neg 0 = 1$.

$\Rightarrow$ $f(\underline{x})$ can be built from "and", "or", "not" gates, and copy gates $x \mapsto (x, x)$.

"Universal gate set"

(Note: In fact, "nand" $\neg(x \wedge y)$ or "nor" $\neg(x \vee y)$ are by themselves already universal together w/ copy.)

"Circuit model" of computation: $f \equiv f^{(n)}$ can be built from a simple universal gate set w/out loops (i.e., gates are applied sequentially).
(Technical point: The circuit family for $n \in \mathbb{N}$ must be generatable in an efficient way, e.g. a simple & fast program.)

The <u>hardness</u> of a problem is meas. by number $K(n)$ gates needed to compute $f^{(n)}$. ($\hat{=}$ # of time steps).

Can distinguish different regimes:

$K(n) \sim \mathrm{poly}(n)$: efficiently solvable (<u>class P</u>)

$K(n) \gtrsim \mathrm{poly}(n)$, e.g. $K(n) \sim \exp(n)$: hard problem

<u>Are there more hard or easy problems?</u>

random $f : \{0,1\}^n \longrightarrow \{0,1\}$

# of possible f : $\quad 2^{(2^n)}$

2^{n}: # of inputs

2: 0 or 1 for each input

But there are only $c^{\mathrm{poly}(n)}$ circuits of length $\mathrm{poly}(n)$!

c: # univ. gates

$\Longrightarrow$ most f cannot be computed efficiently!

Does comp. power (= what is easy) <u>dep. on gate set</u>?

$\longrightarrow$ No. By def., any univ. gate set can simulate any other gate set of few-qubit gates w/ constant overhead!

Are there other models of computation?

- CPU + RAM
- parallel computer
- "Turing machine" (tape + head ≡ "linear RAM")
- cellular automata

⋮

But: All known "reasonable" models of computation can simulate each other w/ poly(n) overhead ⇒ same comput. power!

<u>Church-Turing-Thesis</u>: All reasonable models of computation have the same computational power.

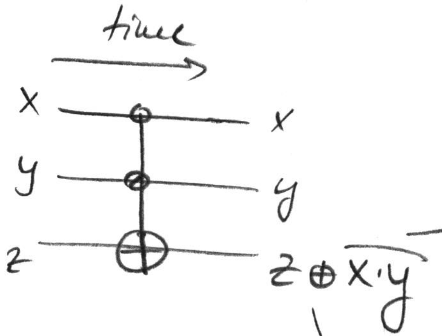
We will use <u>circuit model</u> for quantum computers: Gates become <u>Unitaries</u>!

<u>But</u>: Unitaries reversible ⟷ class. gates irreversible

So... can we even fit class. computation into that?

Yes! – Class. computation can be turned reversible:

Toffoli gate:



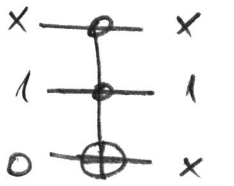
"and"

"XOR": $0 \oplus 0 = 0$
(= addition mod 2) $1 \oplus 1 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$

⟶ reversible!

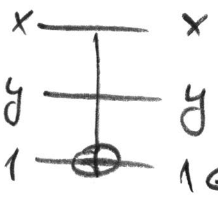
⟶ can simulate and, or, not, copy using ancillas,

e.g.:



COPY

;



$1 \oplus (x \cdot y) = \neg(x \cdot y)$: NAND

⟹ reversible univ. gate set (using ancillas)

Any $f(\underline{x})$ can be computed reversibly:

bitwise XOR?

$$f^r(\underline{x}, y) = (\underline{x}, f(\underline{x}) \oplus y)$$

(Possible w/ ancillas w/ ess. the same circuit.

Idea: Comp. f rev. w/ ancillas, XOR result w/ y,

and "un-compute" everything, i.e. reset ancillas.
Can be optimized to use few ancillas → Preskill's notes.)

→ Everything can be computed reversibly.

<u>But</u> 3-bit gate needed!

<u>b) Quantum Circuits</u>

Model of quantum computation – circuit model:

- System consists of qubits: tensor prod. structure
- Universal gate set $S = \{U_1, \dots, U_k\}$ of "small" (i.e., few-qubit) gates
- build circuits
- Input: <u>Classical</u> input $|x_1\rangle|x_2\rangle \dots |x_n\rangle \equiv |x_1, x_2, \dots\rangle$ in <u>computational basis</u> $\{|0\rangle, |1\rangle\}$, and possibly <u>ancillas</u> in $|0\rangle$ state.
- Output: Measure some (or all) qubits at end in comp. basis

(<u>Notes</u>: ◦ Other meas., e.g. POVM, can be simulated w/in model.
◦ Meas. at earlier time can be always postponed.)

Which gate set should we choose?

- Continuum of gates: much more rich!
- Different notions of universality
  - exact universality: any n-qubit gate U can be realized exactly
  - approximate universality: any n-qubit gate U can be approximated by gate set well. (Solovay-Kitaev Thm: ε-approx. of n-qubit gate requires $O(\mathrm{poly}(\log(1/\varepsilon)))$ gates.)
- Turns out: 1 & 2-qubit gates alone univ. ($\leftrightarrow$ classical: 3)
- Examples of approx. univ. gate sets
  - any random 2-qubit gate (!)
  - more later, after introducing some std. gates.

- Our exact universal gate set:

(i) 1-qubit rotations about X & Z axis:

$$R_X(\phi) = e^{-iX\phi/2} \; ; \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}; \; X^2 = I$$

$$R_Z(\phi) = e^{-iZ\phi/2} \; ; \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}; \; Z^2 = I$$

If $\Pi^2 = I$: $e^{-i\Pi\phi/2} = \cos\phi/2 \; I - i \sin\phi/2 \; \Pi$

$$\Rightarrow \quad R_x(\phi) = \begin{pmatrix} \cos\phi/2 & -i\sin\phi/2 \\ -i\sin\phi/2 & \cos\phi/2 \end{pmatrix}$$

$$R_z(\phi) = \begin{pmatrix} e^{-i\phi/2} & 0 \\ 0 & e^{i\phi/2} \end{pmatrix}$$

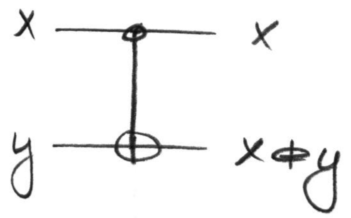
Can also be interpreted as rotation on Bloch sphere, i.e. in $O(3) \cong SU(2)/\mathbb{Z}_2$, about X & Z by ϕ.

R_x & R_z can generate any rotation in $SU(2)/\mathbb{Z}_2 \cong O(3)$ (→ Euler angles!)

$$U = R_x(\alpha) R_z(\beta) R_x(\gamma) \; : \; \text{all of } SU(2).$$

(ii) one 2-qubit gate (almost all would do!)

Typ. we choose "controlled-NOT" ≡ CNOT

CNOT = 

$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

CNOT flips y iff $x = 1$: classical gate!

Can show: This gate set can create any n-qubit U exactly (of course not efficiently, e.g. by parameter counting).

Some important gates + identities ($\to$ HW!)

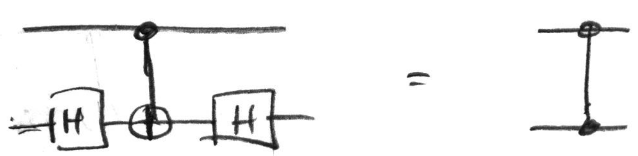
Hadamard gate: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$; $H = H^\dagger$, $H^2 = I$

$$H R_x(\phi) H = R_z(\phi)$$

$$H R_z(\phi) H = R_x(\phi)$$

Graphical notation:

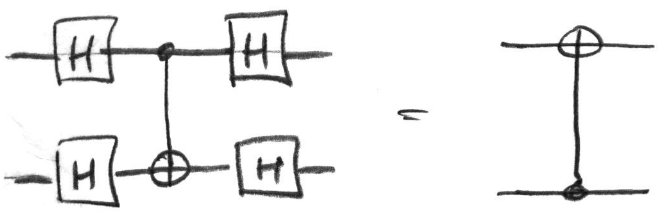
$(X = R_x(\pi), Z = R_z(\pi))$



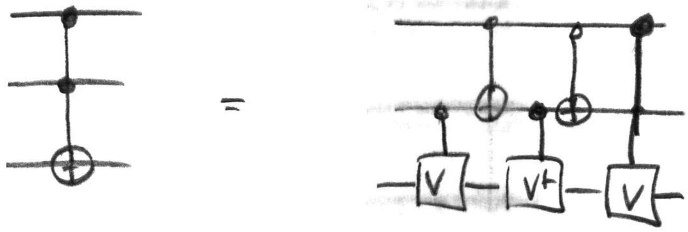
$= \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix}$

"Controlled-Z", CZ

"Controlled phase", CPHASE



Toffoli:

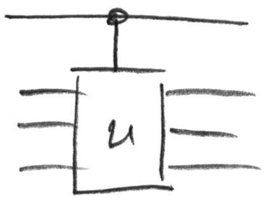


with $V = \frac{1-i}{2}(I + iX)$, and

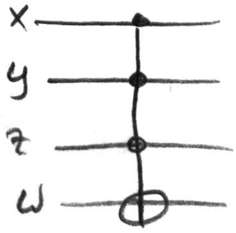
$= \left(\begin{array}{cc|c} 1 & & 0 \\ & 1 & \\ \hline 0 & & V \end{array}\right)$

"controlled-V"

If we can build a classical U, we can also build "controlled-U":



just replace every Toffoli (class. universal!) by Toffoli w/ 3 controls:



(flip w iff $x = y = z = 1$)

Can be built from normal Toffoli ($\to$ HW)

Some more approx. univ. gate sets:

- CNOT + 2 random 1-qubit gates
- CNOT + H + $T \equiv R_Z(\pi/4)$ ("$\pi/8$ gate")