Quantum Computation (Preskill ch. 6)

Quantum Computation

Classical Circuits

- Universal Gates
- Circuit Complexity

Quantum Circuits

- Quantum Complexity
- Universal Quantum Gates

Review of Classical Circuit Theory

Think of a Computation as a function that maps *n* bits to *m* bits

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

Maps *n* bits to *m* bits

A function with an *m* bit output is equivalent to *m* functions with a *one* bit output, so the basic task can be broken into *m* functions mapping *n* bits to *one* bit

There are 2^n possible inputs w/2 possible outputs, so a total of 2^{2^n} functions that map *n* bits to *one* bit

$$\boxed{f : \{0,1\}^n \to \{0,1\}}$$

2^{2^n} of these simple functions

Function evaluation ⟷ sequence of logic operations

Given a binary input $x = x_1 x_2 \ldots x_n$

⇨ separate in sets $\begin{cases} f(x) = 1 \\ f(x) = 0 \end{cases}$

Consider the input

$x^{(a)} : f(x^{(a)}) = 1$ ⇨ define $f^{(a)}(x) = \begin{cases} 1 & \text{for } x = x^{(a)} \\ 0 & \text{for } x \neq x^{(a)} \end{cases}$

one of the *m* simple functions

n of these

Given, for example, we implement $f^{(a)}$ w/logic operations

$$x = \begin{cases} 111\ldots \to f(x) = x_1 \wedge x_2 \wedge x_3 \ldots \wedge x_n \\ 0110\ldots \to f(x) = (\neg x_1) \wedge x_2 \wedge x_3 \wedge (\neg x_4) \ldots \end{cases}$$

Finally, given the $f^{(a)}(x)$'s we can implement the $f(x)$'s as

$$\boxed{f(x) = f^{(1)}(x) \vee f^{(2)}(x) \vee \ldots \vee f^{(n)}(x)}$$

Quantum Computation (Preskill ch. 6)

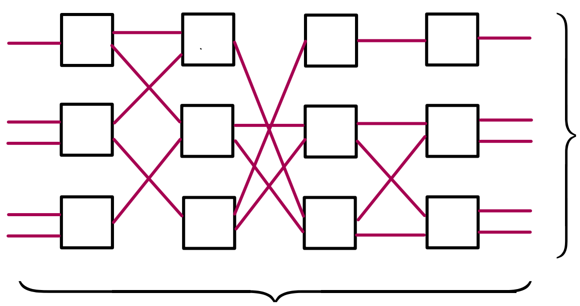
Circuit Complexity

(Pick a universal gate set)

Central Question: How hard is it to solve **PROBLEM** ?

* One measure is the size of the smallest circuit that solves it

Size = Width x Depth



Width

Depth

Consider a circuit family $\{C_n\}$ that solves a decision problem $f: \{0,1\}^n \to \{0,1\}$

Examples

FACTORING $f(x,y) = \begin{cases} 1 & \text{if integer } x \text{ has divisor } < y \\ 0 & \text{otherwise} \end{cases}$

HAMILTONIAN PATH $f(x,y) = \begin{cases} 1 & \text{if graph } x \text{ has Hamiltonian Path} \\ 0 & \text{otherwise} \end{cases}$

We define:

Easy Problems:	$\text{size}(C_n) \leq \text{poly}(n)$
Hard Problems:	$\text{size}(C_n) > \text{poly}(n)$

This distinction allows us to define Complexity Classes, for example

Problem Class P = { Decision Problems solved by a polynomial-sized circuit }

Quantum Computation (Preskill ch. 6)

Consider a circuit family $\{C_n\}$ that solves a decision problem $\quad f: \{0,1\}^n \to \{0,1\}$

Examples

FACTORING $\quad f(x,y) = \begin{cases} 1 & \text{if integer } x \text{ has divisor } < y \\ 0 & \text{otherwise} \end{cases}$

HAMILTONIAN PATH $\quad f(x,y) = \begin{cases} 1 & \text{if graph } x \text{ has Hamiltonian Path} \\ 0 & \text{otherwise} \end{cases}$

We define:

Easy Problems:	$\text{size}(C_n) \le \text{poly}(n)$
Hard Problems:	$\text{size}(C_n) > \text{poly}(n)$

This distinction allows us to define Complexity Classes, for example

Problem Class P = { Decision Problems solved by polynomial-sized circuit }

* Whether PROBLEM $\in P$ is independent of circuit design, universal gate set & other specifics

* Problems in P are special – they have structure that allows efficient computation

Note: The majority of functions $\notin P$

For example, if the output $f(x) \sim$ random we must compute $f(x)$ by lookup table with 2^n entries

Circuit that does lookup has exponential size

Special Class: One-Way Function

Problem Class NP = { PROBLEM is easy or hard, but the answer is easy to check }

Stands for "Non-deterministic Polynomial Time

Examples: FACTORING $\in$ NP

HAMILTONIAN PATH $\in$ NP

Note: Clearly $P \subseteq NP$, Conjecture that $P \neq NP$

Quantum Computation (Preskill ch. 6)

- ✱ Whether PROBLEM $\in \mathcal{P}$ is independent of circuit design, universal gate set & other specifics
- ✱ Problems in $\mathcal{P}$ are special – they have structure that allows efficient computation

Note: The majority of functions $\notin \mathcal{P}$

For example, if the output $f(x) \sim$ random we must compute $f(x)$ by lookup table with 2^n entries

Circuit that does lookup has exponential size

Special Class:

One-Way Function

Problem Class NP = $\left\{ \begin{array}{c} \text{PROBLEM is easy or hard, but} \\ \text{the answer is easy to check} \end{array} \right\}$

Stands for "Non-deterministic Polynomial Time

Examples: FACTORING $\in$ NP

HAMILTONIAN PATH $\in$ NP

Note: Clearly $P \subseteq NP$, Conjecture that $P \neq NP$

Special Problem: CIRCUIT-SAT $\in$ NP

Input = Circuit w/n gates, m input bits

Problem = is there an m-bit input w/output = 1

$$f(C) = \begin{cases} 1 & \text{if } \exists\, x^{(m)} \text{ so } C(x^{(m)}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Easy to check solution because if we have the input circuit C we can run it with the input $x^{(m)}$ and determine if it evaluates to 1.

Cooks Theorem: Every PROBLEM $\in$ NP is polynomially reducible to CIRCUIT-SAT

$\left.\begin{array}{c} \text{CIRCUIT-SAT} \\ \text{HAMILTONIAN PATH} \end{array}\right\} \in \boxed{\text{NP- Complete}}$ NPC $\neq$ NP

Quantum Computation (Preskill ch. 6)

Special Problem: CIRCUIT-SAT $\in$ NP

Input = Circuit w/n gates, m input bits

Problem = is there an m-bit input w/output = 1

$$f(C) = \begin{cases} 1 & \text{if } \exists\, x^{(m)} \text{ so } C(x^{(m)}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Easy to check solution because if we have the input circuit C we can run it with the input $x^{(m)}$ and determine if it evaluates to 1.

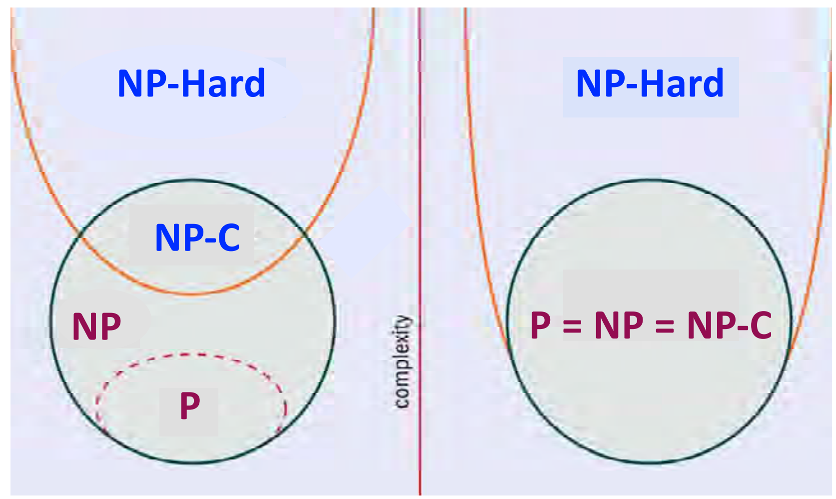
Cooks Theorem: Every PROBLEM $\in$ NP is polynomially reducible to CIRCUIT-SAT

$$\left.\begin{array}{c}\text{CIRCUIT-SAT} \\ \text{HAMILTONIAN PATH}\end{array}\right\} \in \boxed{\text{NP- Complete}}$$

NPC ≠ NP

Complexity Hierarchy

* **Conjecture**: P $\in$ NP
* $\exists$ Problems in NP that are neither P or NPC
* NPI: Problems of intermediate difficulty
* Conjecture: Factoring $\in$ NPI



P ≠ NP

P = NP

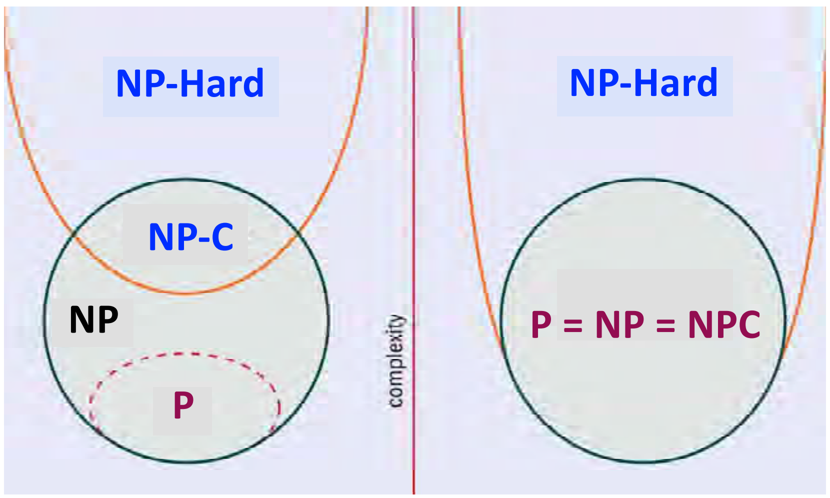
Takeaway Message

* Complexity theory is a rich field with many known complexity classes
* Many foundational conjectures remain unproven
* As we will see, switching to Quantum Circuits changes things

Quantum Computation (Preskill ch. 6)

Complexity Hierarchy

- **Conjecture**: P ∈ NP
- ∃ Problems in NP that are neither P or NPC
- NPI: Problems of intermediate difficulty
- Conjecture: Factoring ∈ NPI



P ≠ NP   P = NP

Takeaway Message

- Complexity theory is a rich field with many known complexity classes
- Many foundational conjectures remain unproven
- As we will see, switching to Quantum Circuits changes things

Aside: Classical Reversible Computation

Motivation:

Quantum Computation = Unitary Transformation

⇓

Reversible !

Classical Reversible Comp: $f: \{0,1\}^n \to \{0,1\}^n$

Repackage $f: \{0,1\}^n \to \{0,1\}^m$ as reversible

$$f: \{0,1\}^{n+m} \to \{0,1\}^{n+m}$$
$$f(x, 0^{(m)}) = (x; f(x))$$

we separate $n + m$ qubit register into input and output so no information is lost

Note: Not all 1 & 2-bit gates are reversible, e. g., AND, OR, ERASE

Quantum Computation (Preskill ch. 6)

Quantum Circuits

Classical Computer = finite set of gates acting on bits

Quantum Computer = finite set of <u>quantum gates</u> acting on <u>quantum bits</u>

Quantum Computation:

$$U : |000\ldots0\rangle \rightarrow |\psi\rangle$$

unitary composed of finite # of gates

n qubit input

output = outcome of Orthog. Measurement in basis $\{|0\rangle, |1\rangle\}^n$

Note:

* The Hilbert space of the Quantum Computer has a preferred decomposition into tensor producs of low dimensional spaces (qubits), respected by gates which act on only a few qubits at a time.

  - This helps establish notion of Quantum Complexity

* Decomposition into subsystems and local manipulations means gates act on qubits in a bounded region.

* It is suspected, but not proven, that the power of Q. C. derives from this decomposition:

$$\boxed{n \text{ qubits} \rightarrow 2^n \text{ dimensional } \mathcal{H} \text{ resource grows } \sim 2^n}$$