Lecture 12 - Circuits

TMe = "Software"
Circuits = "hardware"

- Can always unravel software running in time $T$ into hardware of size $T^{o(1)}$ ("software" vs. "hardware")

Q: Is hardware of size poly(n) more powerful than software running in time poly(n) for problems of size $n$?

Ans: "Technically yes, but basically no (we think)."

**Def:** A circuit $C: \{0,1\}^n \rightarrow \{0,1\}$ is a DAG with vertices that are variables ($x_1, x_2, \ldots, x_n$) or gates ($\land$, $\lor$, $\neg$).

- $\text{size}(C) = \# \text{ gates}$
- By specifying an output gate, $C$ defines a function from $\{0,1\}^n$ to $\{0,1\}$
- $\text{size}(f) = \text{size of the smallest circuit implementing } f.$

**Eq:** $\{0,1\}^2 \rightarrow \{0,1\}$

\[ \land : \{0,1\}^2 \rightarrow \{0,1\} \]
\[ \lor : \{0,1\}^2 \rightarrow \{0,1\} \]
\[ \neg : \{0,1\} \rightarrow \{0,1\} \]

For any $f : \{0,1\}^n \rightarrow \{0,1\}$, $\text{size}(f) = O(n2^n)$ by encoding its truth table:

$$f(x) = \lor \land (x=y_1) = \lor (\forall y \in \{0,1\}^n : (x,y) \land Eq(x,y) \land \neg \land Eq(x,y_1))$$
Q: What if we used a different gate set?

Let \( G = \{ g : \{0,1\}^k \rightarrow \{0,1\}^k \text{ for some } k = O(1) \}, k \geq 2 \).

\[ \text{Size}(G) = \Theta(\text{Size}_{\text{nnv}}(G)) \]

\[ p^2 : \text{Size}_{\text{nnv}}(g) \leq O(k2^k) = O(1). \]

Let \( f^m_L : \{0,1\}^m \rightarrow \{0,1\} \) be the indicator function

\[ f^m_L(x) := \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{if } x \notin L \end{cases} \]

def: \( \text{SIZE}(\text{nnv}) = \{ L : \text{Size}(f^m_L) = O(m) \} \)

\( \text{P/poly} := \text{SIZE}(\text{poly} \times \text{nnv}) \)

Q: P vs P/poly?

Q: NP vs P/poly?
\[ \text{TIME}(t(n)) \leq \text{SIZE}(s(n)^2) \quad \text{(in fact } \leq \text{SIZE}(s\log s)) \]

Cor. \( P \subseteq \text{P/poly} \).

\[ \text{pf.: Let } L \in \text{TIME}(t(n)); \text{ assume } \Sigma_L = \{0,1\} \text{ for simplicity.} \]

Let \( M \) be an oblivious TM for \( L \) running in time \( O(t(n)^2) \).

\( M \) defined by

\[ S : Q \times \{0,1\}^* \rightarrow Q \]

\[ \delta : Q \times \{0,1\}^* \rightarrow \mathbb{Q}, \{0,1\} \]

Encode \( Q \) as \( f_0, f_1 \).

Encode \( \text{Accept} \) as \( 1_q \), \( \text{Start} \) as \( 0_q \).

Since \( q = \log_2(1) = 0(1) \),

\[ \text{size}(S) \leq O(1), \]

\[ \text{size}(\delta) \leq O(1) \]

\[ \Rightarrow \text{size}(C) \leq O(t(n)^2) \]
def: For $L \subseteq \Sigma^*$, we define the unary version of $L$

$L_u : = \{1^n : (n \text{ in binary}) \in L \}$.

* For any $L$, $L_u \in P/poly$.

pf: We use the advice function

$$a(n) = \begin{cases} 1 & \text{if } 1^n \in L_u \\ 0 & \text{if } 1^n \notin L_u \end{cases}$$

Our TM $M(x, a)$ just checks if $x$ is of the form $1^n$ and outputs $a(n)$.

Cor: $P \neq P/poly$

pf: Let $L$ be the Halting problem. Then

$L_u \in P/poly$

but

$L \notin P/poly$
def: Let \( a : \mathbb{N} \to \{0,1\}^* \) be an "advice function".

A TM with advice \((M, a)\) computes the function

\[ x \mapsto M(x, a(|x|)). \]

\( \text{P/poly (advice)} := \{ L : L \text{ can be decided by a TM with advice running in poly(n) time} \} \)

\( \text{P/poly (advice)} = \text{P/poly (circuit)} \)

\( \text{P/poly} \quad \exists : \quad \text{give the circuit as advice and eval on x.} \)

\( \leq : \quad \text{for a fixed input length n, implement} \ M(x,a) \ \text{as a circuit} \ \langle x, a \rangle, \) 

\( \text{Hardwire} \ \alpha(\mathbb{N}) \ \text{into} \ C. \)

Perspective on \( \text{P/poly vs NP} \):

\( \text{P/poly} = \text{"trusted advice, one per input size"} \)

\( \text{NP} = \text{"untrusted advice, one per input" (have to certify.)} \)
* Circuit-SAT is NP-hard.

pf: Let $F$ be the indicator function of some language $L \in \text{NP}$. Then

$$F(x) = \bigvee_{y \in \text{poly}(x)} F(x, y)$$

for some poly-time function $F$.

$$= \bigvee_{y \in \text{poly}(x)} C(x, y)$$

for some poly-size circuit $C$.

So $F(x) = 1$ iff $C_x(y) := C(x, y)$ is satisfiable.

* 3-SAT is NP-hard.

Circuit $f(x)$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
</table>

$$f(x, y) = (\bigwedge_i C_i) \lor y_j$$

$f(x) = 1$ iff $F(x, y)$ is satisfiable in $y$.

(In fact, $F(x, y)$ is uniquely satisfiable).

$$g_1 = x_1 \land x_2$$

$$g_2 = g_1 \lor x_3$$

$$g_3 = g_2 \lor g_k$$

$$g_{j+1} = g_j \lor g_k$$

$$g_k = g_{k+1}$$

Any function on 3 vars can be implemented as a 3-CNF (write it as a 3-DNF).
99% of functions require exponential-size circuits (and therefore also polynomial time to solve).

Proof: Consider functions \( f : \{0,1\}^n \rightarrow \{0,1\}^n \) (there are \( 2^{2^n} \)).

Let \( \mathcal{F}(s) = \{ f : \text{size}(f) \leq s \} \).

Then \( |\mathcal{F}(s)| \leq (2s + 2s^2)^n \leq 2^{c \log s} \).

So the fraction of functions with size \( \leq s \) is

\[
\frac{|\mathcal{F}(s)|}{2^{2^n}} \leq \frac{2^{c \log s}}{2^{2^n}} = 2^{c \log s - 2^n}.
\]

Solving \( 2^{c \log s - 2^n} = \frac{1}{100} \) gives

\[
s \geq A(2^n) \geq A\left(2^{\log 2^n} \right).
\]

Open Problem: Prove that some explicit function requires circuits of size \( \geq 8n \).