# Lecture 30

## Meyer's Theorem, Circuit Lower Bound

# Meyer’s Theorem

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:**

# Meyer's Theorem

**Theorem:** If EXP $\subseteq$ $\mathsf{P}_{/\mathsf{poly}}$, then EXP $= \Sigma_2^p$.

**Proof:** Let $L \in$ EXP and $M$ be an oblivious TM that decides $L$ in $O(2^{n^c})$ time.

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Let $L \in \mathsf{EXP}$ and $M$ be an oblivious TM that decides $L$ in $O(2^{n^c})$ time.

$$x \in L \iff \exists \text{ a sequence of snapshots } T_1, T_2, \ldots, T_{O(2^{n^c})} \text{ such that } \ldots\ldots$$

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Let $L \in \mathsf{EXP}$ and $M$ be an oblivious TM that decides $L$ in $O(2^{n^c})$ time.

$$x \in L \iff \exists \text{ a sequence of snapshots } T_1, T_2, \ldots, T_{O(2^{n^c})} \text{ such that } \ldots\ldots$$

Define a related language $L'$

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Let $L \in \mathsf{EXP}$ and $M$ be an oblivious TM that decides $L$ in $O(2^{n^c})$ time.

$x \in L \iff \exists$ a sequence of snapshots $T_1, T_2, \ldots, T_{O(2^{n^c})}$ such that ……

Define a related language $L'$

$L' = \{(x, i, j) \mid j$th bit of the $i$th snapshot is $1$ when $M$ runs on $x\}$

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Let $L \in \mathsf{EXP}$ and $M$ be an oblivious TM that decides $L$ in $O(2^{n^c})$ time.

$$x \in L \iff \exists \text{ a sequence of snapshots } T_1, T_2, \ldots, T_{O(2^{n^c})} \text{ such that } \ldots\ldots$$

Define a related language $L'$

$$L' = \{(x, i, j) \mid j\text{th bit of the } i\text{th snapshot is } 1 \text{ when } M \text{ runs on } x\}$$

Now,

# Meyer's Theorem

**Theorem:** If EXP $\subseteq$ P$_{\text{/poly}}$, then EXP $= \Sigma_2^p$.

**Proof:** Let $L \in$ EXP and $M$ be an oblivious TM that decides $L$ in $O(2^{n^c})$ time.

$$x \in L \iff \exists \text{ a sequence of snapshots } T_1, T_2, \ldots, T_{O(2^{n^c})} \text{ such that } \ldots\ldots$$

Define a related language $L'$

$$L' = \{(x, i, j) \mid j\text{th bit of the } i\text{th snapshot is } 1 \text{ when } M \text{ runs on } x\}$$

Now,

$L \in$ EXP

# Meyer's Theorem

**Theorem:** If EXP $\subseteq$ P$_{/\text{poly}}$, then EXP $= \Sigma_2^p$.

**Proof:** Let $L \in$ EXP and $M$ be an oblivious TM that decides $L$ in $O(2^{n^c})$ time.

$$x \in L \iff \exists \text{ a sequence of snapshots } T_1, T_2, \ldots, T_{O(2^{n^c})} \text{ such that } \ldots\ldots$$

Define a related language $L'$

$$L' = \{(x, i, j) \mid j\text{th bit of the } i\text{th snapshot is } 1 \text{ when } M \text{ runs on } x\}$$

Now,

$$L \in \textsf{EXP} \implies L' \in \textsf{EXP}$$

# Meyer's Theorem

**Theorem:** If EXP $\subseteq$ P$_{/\text{poly}}$, then EXP $= \Sigma_2^p$.

**Proof:** Let $L \in$ EXP and $M$ be an oblivious TM that decides $L$ in $O(2^{n^c})$ time.

$$x \in L \iff \exists \text{ a sequence of snapshots } T_1, T_2, \ldots, T_{O(2^{n^c})} \text{ such that } \ldots\ldots$$

Define a related language $L'$

$$L' = \{(x, i, j) \mid j\text{th bit of the } i\text{th snapshot is } 1 \text{ when } M \text{ runs on } x\}$$

Now,

$$L \in \text{EXP} \implies L' \in \text{EXP} \implies L' \in \text{P}_{/\text{poly}}$$

# Meyer's Theorem

**Theorem:** If **EXP** $\subseteq$ **P**$_{\mathbf{/poly}}$, then **EXP** $= \Sigma_2^p$.

**Proof:** Let $L \in$ **EXP** and $M$ be an oblivious TM that decides $L$ in $O(2^{n^c})$ time.

$$x \in L \iff \exists \text{ a sequence of snapshots } T_1, T_2, \ldots, T_{O(2^{n^c})} \text{ such that } \ldots\ldots$$

Define a related language $L'$

$$L' = \{(x, i, j) \mid j\text{th bit of the } i\text{th snapshot is } 1 \text{ when } M \text{ runs on } x\}$$

Now,

$$L \in \mathbf{EXP} \implies L' \in \mathbf{EXP} \implies L' \in \mathbf{P}_{\mathbf{/poly}} \implies$$

# Meyer's Theorem

**Theorem:** If EXP $\subseteq$ P$_{/\text{poly}}$, then EXP $= \Sigma_2^p$.

**Proof:** Let $L \in$ EXP and $M$ be an oblivious TM that decides $L$ in $O(2^{n^c})$ time.

$$x \in L \iff \exists \text{ a sequence of snapshots } T_1, T_2, \ldots, T_{O(2^{n^c})} \text{ such that } \ldots\ldots$$

Define a related language $L'$

$$L' = \{(x, i, j) \mid j\text{th bit of the } i\text{th snapshot is } 1 \text{ when } M \text{ runs on } x\}$$

Now,

$$L \in \text{EXP} \implies L' \in \text{EXP} \implies L' \in \text{P}_{/\text{poly}} \implies$$ $\exists$ a polysize circuit family $C$ that on input $(x, i, j)$ outputs the $j$th bit of $i$th snapshot of $M$'s run on $x$.

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Let $L \in \mathsf{EXP}$ and $M$ be an oblivious TM that decides $L$ in $O(2^{n^c})$ time.

$$x \in L \iff \exists \text{ a sequence of snapshots } T_1, T_2, \ldots, T_{O(2^{n^c})} \text{ such that } \ldots\ldots$$

Define a related language $L'$

$$L' = \{(x, i, j) \mid j\text{th bit of the } i\text{th snapshot is } 1 \text{ when } M \text{ runs on } x\}$$

Now,

$$L \in \mathsf{EXP} \implies L' \in \mathsf{EXP} \implies L' \in \mathsf{P}_{/\mathsf{poly}} \implies$$

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Let $L \in \mathsf{EXP}$ and $M$ be an oblivious TM that decides $L$ in $O(2^{n^c})$ time.

$$x \in L \iff \exists \text{ a sequence of snapshots } T_1, T_2, \ldots, T_{O(2^{n^c})} \text{ such that } \ldots\ldots$$

Define a related language $L'$

$$L' = \{(x, i, j) \mid j\text{th bit of the } i\text{th snapshot is } 1 \text{ when } M \text{ runs on } x\}$$

Now,

$L \in \mathsf{EXP} \implies L' \in \mathsf{EXP} \implies L' \in \mathsf{P}_{/\mathsf{poly}} \implies$ $\exists$ a polysize circuit family $D$ that on input $(x, i)$ outputs the $i$th snapshot of $M$'s run on $x$.

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:**

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$$x \in L$$

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$$x \in L \quad \Longleftrightarrow$$

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$\exists$ a circuit $C$

$$x \in L \quad \Longleftrightarrow$$

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$$\exists \text{ a circuit } C \text{ s.t. } \forall i \in [1, O(2^{n^c})]$$

$$x \in L \quad \Longleftrightarrow$$

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$\exists$ a circuit $C$ s.t. $\forall i \in [1, O(2^{n^c})]$ the following are true:

$x \in L \iff$

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$\exists$ a circuit $C$ s.t. $\forall i \in [1, O(2^{n^c})]$ the following are true:

$x \in L \iff$ 1) If $i = 1$, then $C(x, i)$ is the starting snapshot of $M$'s run on $x$.

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$\exists$ a circuit $C$ s.t. $\forall i \in [1, O(2^{n^c})]$ the following are true:

$x \in L \iff$

1) If $i = 1$, then $C(x, i)$ is the starting snapshot of $M$'s run on $x$.

2) If $i = O(2^{n^c})$, then $C(x, i)$ is the accepting snapshot of $M$'s run on $x$.

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$\exists$ a circuit $C$ s.t. $\forall i \in [1, O(2^{n^c})]$ the following are true:

$x \in L \iff$

1) If $i = 1$, then $C(x, i)$ is the starting snapshot of $M$'s run on $x$.

2) If $i = O(2^{n^c})$, then $C(x, i)$ is the accepting snapshot of $M$'s run on $x$.

3) If $i > 1$, $C(x, i) = F(C(x, i-1), x_{inputpos(i)}, C(x, prev(i)))$

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$\exists$ a circuit $C$ s.t. $\forall i \in [1, O(2^{n^c})]$ the following are true:

$x \in L \iff$

1) If $i = 1$, then $C(x, i)$ is the starting snapshot of $M$'s run on $x$.

2) If $i = O(2^{n^c})$, then $C(x, i)$ is the accepting snapshot of $M$'s run on $x$.

3) If $i > 1$, $C(x, i) = F(C(x, i-1), x_{inputpos(i)}, C(x, prev(i)))$

Polytime TM $N$ exists that on input $(x, C, i)$

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$\exists$ a circuit $C$ s.t. $\forall i \in [1, O(2^{n^c})]$ the following are true:

$x \in L \iff$

1) If $i = 1$, then $C(x, i)$ is the starting snapshot of $M$'s run on $x$.

2) If $i = O(2^{n^c})$, then $C(x, i)$ is the accepting snapshot of $M$'s run on $x$.

3) If $i > 1$, $C(x, i) = F(C(x, i-1), x_{inputpos(i)}, C(x, prev(i)))$

Polytime TM $N$ exists that on input $(x, C, i)$ outputs $1$ iff

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$\exists$ a circuit $C$ s.t. $\forall i \in [1, O(2^{n^c})]$ the following are true:

$x \in L \iff$

1) If $i = 1$, then $C(x, i)$ is the starting snapshot of $M$'s run on $x$.

2) If $i = O(2^{n^c})$, then $C(x, i)$ is the accepting snapshot of $M$'s run on $x$.

3) If $i > 1$, $C(x, i) = F(C(x, i-1), x_{inputpos(i)}, C(x, prev(i)))$

Polytime TM $N$ exists that on input $(x, C, i)$ outputs $1$ iff conditions $1)$, $2)$ and $3)$ are met.

# Meyer's Theorem

**Theorem:** If **EXP** $\subseteq$ **P$_{\text{/poly}}$**, then **EXP** $= \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$x \in L \iff$

$\exists$ a circuit $C$ s.t. $\forall i \in [1, O(2^{n^c})]$ the following are true:

1) If $i = 1$, then $C(x, i)$ is the starting snapshot of $M$'s run on $x$.

2) If $i = O(2^{n^c})$, then $C(x, i)$ is the accepting snapshot of $M$'s run on $x$.

3) If $i > 1$, $C(x, i) = F(C(x, i-1), x_{inputpos(i)}, C(x, prev(i)))$

Polytime TM $N$ exists that on input $(x, C, i)$ outputs $1$ iff conditions $1)$, $2)$ and $3)$ are met.

**Q:** How will $N$ calculate $inputpos(i)$ and $prev(i)$ in polytime?

# Meyer's Theorem

**Theorem:** If **EXP** $\subseteq$ **P/poly**, then **EXP** $= \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$x \in L \iff$ $\exists$ a circuit $C$ s.t. $\forall i \in [1, O(2^{n^c})]$ the following are true:

1) If $i = 1$, then $C(x, i)$ is the starting snapshot of $M$'s run on $x$.

2) If $i = O(2^{n^c})$, then $C(x, i)$ is the accepting snapshot of $M$'s run on $x$.

3) If $i > 1$, $C(x, i) = F(C(x, i-1), x_{inputpos(i)}, C(x, prev(i)))$

Polytime TM $N$ exists that on input $(x, C, i)$ outputs $1$ iff conditions $1)$, $2)$ and $3)$ are met.

**Q:** How will $N$ calculate $inputpos(i)$ and $prev(i)$ in polytime?

Oblivious TMs follow a pattern in head movement that can be used to compute

# Meyer's Theorem

**Theorem:** If $\mathsf{EXP} \subseteq \mathsf{P}_{/\mathsf{poly}}$, then $\mathsf{EXP} = \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$\exists$ a circuit $C$ s.t. $\forall i \in [1, O(2^{n^c})]$ the following are true:

$x \in L \iff$

1) If $i = 1$, then $C(x, i)$ is the starting snapshot of $M$'s run on $x$.

2) If $i = O(2^{n^c})$, then $C(x, i)$ is the accepting snapshot of $M$'s run on $x$.

3) If $i > 1$, $C(x, i) = F(C(x, i-1), x_{inputpos(i)}, C(x, prev(i)))$

Polytime TM $N$ exists that on input $(x, C, i)$ outputs $1$ iff conditions $1$), $2$) and $3$) are met.

**Q:** How will $N$ calculate $inputpos(i)$ and $prev(i)$ in polytime?

Oblivious TMs follow a pattern in head movement that can be used to compute $inputpos(i)$ and $prev(i)$ in polytime.

# Meyer's Theorem

**Theorem:** If EXP $\subseteq$ P$_{/\text{poly}}$, then EXP $= \Sigma_2^p$.

**Proof:** Putting $L$ in $\Sigma_2^p$:

$x \in L \iff$ $\exists$ a circuit $C$ s.t. $\forall i \in [1, O(2^{n^c})]$ the following are true:

1) If $i = 1$, then $C(x, i)$ is the starting snapshot of $M$'s run on $x$.

2) If $i = O(2^{n^c})$, then $C(x, i)$ is the accepting snapshot of $M$'s run on $x$.

3) If $i > 1$, $C(x, i) = F(C(x, i-1), x_{inputpos(i)}, C(x, prev(i)))$

Polytime TM $N$ exists that on input $(x, C, i)$ outputs $1$ iff conditions $1$), $2$) and $3$) are met.

**Q:** How will $N$ calculate $inputpos(i)$ and $prev(i)$ in polytime?

Oblivious TMs follow a pattern in head movement that can be used to compute $inputpos(i)$ and $prev(i)$ in polytime.

$\blacksquare$

# Circuit Lower Bounds

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \rightarrow \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:**

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f : \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$:

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \rightarrow \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$:

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $S$

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f\colon\{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $S$

\# of vertices

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $S \times (2 \log S$

# of vertices

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

vertices giving edges

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $S \times (2 \log S$

# of vertices

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

vertices giving edges

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $S \times (2 \log S + \log S)$

# of vertices

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $\underbrace{S}_{\text{\# of vertices}} \times (\overbrace{2\log S}^{\text{vertices giving edges}} + \underbrace{\log S}_{\text{labels: } x_i,\ \wedge,\ \vee,\ \neg})$

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \rightarrow \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $\underbrace{S}_{\text{\# of vertices}} \times (\overbrace{2 \log S}^{\text{vertices giving edges}} + \underbrace{\log S}_{\text{labels: } x_i,\ \wedge,\ \vee,\ \neg}) \times 3$

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

unique

vertices giving edges

encoding

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $S \times (2\log S + \log S) \times 3$

\# of vertices

labels: $x_i$, $\wedge$, $\vee$, $\neg$

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $\underbrace{S}_{\text{\# of vertices}} \times (\overbrace{2\log S}^{\text{vertices giving edges}} + \underbrace{\log S}_{\text{labels: } x_i, \wedge, \vee, \neg}) \times \overbrace{3}^{\text{unique encoding}}$
- The number of circuits of size $S$ is at most: $2^{9S\log S}$

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $\underbrace{S}_{\text{\# of vertices}} \times (\overbrace{2\log S}^{\text{vertices giving edges}} + \underbrace{\log S}_{\text{labels: } x_i,\ \wedge,\ \vee,\ \neg}) \times \overbrace{3}^{\text{unique encoding}}$
- The number of circuits of size $S$ is at most: $2^{9S\log S}$

Set $S = 2^n/(10n)$.

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

unique

vertices giving edges

encoding

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $S \times (2\log S + \log S) \times 3$
- The number of circuits of size $S$ is at most: $2^{9S\log S}$

\# of vertices

labels: $x_i$, $\wedge$, $\vee$, $\neg$

Set $S = 2^n/(10n)$. Then, the number of circuits of size $S$ is at most:

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $\underbrace{S}_{\text{\# of vertices}} \times (\overbrace{2\log S}^{\text{vertices giving edges}} + \underbrace{\log S}_{\text{labels: } x_i,\ \wedge,\ \vee,\ \neg}) \times \overbrace{3}^{\text{unique encoding}}$
- The number of circuits of size $S$ is at most: $2^{9S\log S}$

Set $S = 2^n/(10n)$. Then, the number of circuits of size $S$ is at most:

$$2^{9S\log S}$$

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f$:$\{0,1\}^n \rightarrow \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

unique

vertices giving edges

encoding

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $S \times (2\log S + \log S) \times 3$
- The number of circuits of size $S$ is at most: $2^{9S\log S}$

# of vertices

labels: $x_i$, $\wedge$ , $\vee$ , $\neg$

Set $S = 2^n/(10n)$. Then, the number of circuits of size $S$ is at most:

$$2^{9S\log S} < 2^{9.2^n/(10n).n}$$

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

unique

vertices giving edges

encoding

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $S \times (2\log S + \log S) \times 3$
- The number of circuits of size $S$ is at most: $2^{9S\log S}$

# of vertices

labels: $x_i$, $\wedge$ , $\vee$ , $\neg$

Set $S = 2^n/(10n)$. Then, the number of circuits of size $S$ is at most:

$$2^{9S\log S} < 2^{9.2^n/(10n).n} < 2^{2^n}$$

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

vertices giving edges

unique encoding

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $S \times (2 \log S + \log S) \times 3$
- The number of circuits of size $S$ is at most: $2^{9S \log S}$

# of vertices

labels: $x_i$, $\wedge$, $\vee$, $\neg$

Set $S = 2^n/(10n)$. Then, the number of circuits of size $S$ is at most:

$$2^{9S \log S} < 2^{9.2^n/(10n).n} < 2^{2^n}$$

The number of functions exceeds the number of circuits.

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f{:}\{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$

vertices giving edges

unique encoding

- The number of bits required to encode a circuit of size $S$: $S \times (2\log S + \log S) \times 3$

# of vertices

labels: $x_i$, $\wedge$, $\vee$, $\neg$

- The number of circuits of size $S$ is at most: $2^{9S\log S}$

Set $S = 2^n/(10n)$. Then, the number of circuits of size $S$ is at most:

$$2^{9S\log S} < 2^{9.2^n/(10n).n} < 2^{2^n}$$

The number of functions exceeds the number of circuits.

$\therefore$ some functions from $\{0,1\}^n$ to $0,1$ cannot be computed by circuits of size $2^n/(10n)$

# Circuit Lower Bounds

**Theorem:** For every $n > 1$, there exists a function $f: \{0,1\}^n \to \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.

**Proof:** Counting argument:

- The number of functions from $\{0,1\}^n$ to $\{0,1\}$: $2^{2^n}$
- The number of bits required to encode a circuit of size $S$: $\underbrace{S}_{\text{\# of vertices}} \times (\overbrace{2\log S}^{\text{vertices giving edges}} + \underbrace{\log S}_{\text{labels: } x_i,\ \wedge,\ \vee,\ \neg}) \times \overbrace{3}^{\text{unique encoding}}$
- The number of circuits of size $S$ is at most: $2^{9S\log S}$

Set $S = 2^n/(10n)$. Then, the number of circuits of size $S$ is at most:

$$2^{9S\log S} < 2^{9.2^n/(10n).n} < 2^{2^n}$$

The number of functions exceeds the number of circuits.

$\therefore$ some functions from $\{0,1\}^n$ to $0,1$ cannot be computed by circuits of size $2^n/(10n)$ ■