HW05 Solution
2) CFG-CN for $L_1 = \{a^{n/4}b^{n/4}a^{n/4}b^{n/4}\}$

CFG with, for each rule, how many rules it becomes in CNF.
2) CFG-CNF for $L_1 = \{a^{n/4}b^{n/4}a^{n/4}b^{n/4}\}$

CFG with, for each rule, how many rules it becomes in CNF.

$S \rightarrow ABAB$. 
2) CFG-CNF for \( L_1 = \{ a^{n/4} b^{n/4} a^{n/4} b^{n/4} \} \)

CFG with, for each rule, how many rules it becomes in CNF. 

\( S \rightarrow ABAB \). CNF: 2 rules.
2) CFG-CNF for \( L_1 = \{a^{n/4}b^{n/4}a^{n/4}b^{n/4}\} \)

CFG with, for each rule, how many rules it becomes in CNF.

\[ S \rightarrow ABAB. \text{ CNF: 2 rules.} \]
\[ A \rightarrow a \cdots a. \text{ } n/4 \text{ a's.} \]
2) CFG-CNF for $L_1 = \{a^{n/4} b^{n/4} a^{n/4} b^{n/4}\}$

CFG with, for each rule, how many rules it becomes in CNF.

$S \rightarrow ABAB$. CNF: 2 rules.

$A \rightarrow a \cdots a$. $n/4$ a’s. CNF: $\log_2(n/4)$ rules.
2) CFG-CNF for \( L_1 = \{ a^{n/4} b^{n/4} a^{n/4} b^{n/4} \} \)

CFG with, for each rule, how many rules it becomes in CNF.

\[ S \rightarrow ABAB. \text{ CNF: } 2 \text{ rules.} \]

\[ A \rightarrow a \cdots a. \ n/4 \ a's. \text{ CNF: } \log_2 (n/4) \text{ rules.} \]

\[ B \rightarrow b \cdots b. \ n/4 \ b's. \]
2) CFG-CNF for $L_1 = \{a^{n/4}b^{n/4}a^{n/4}b^{n/4}\}$

CFG with, for each rule, how many rules it becomes in CNF.

$S \rightarrow ABAB$. CNF: 2 rules.

$A \rightarrow a \cdots a$. $n/4$ $a$’s. CNF: $\log_2(n/4)$ rules.

$B \rightarrow b \cdots b$. $n/4$ $b$’s. CNF: $\log_2(n/4)$ rules.
2) **CFG-CNF for** $L_1 = \{a^{n/4}b^{n/4}a^{n/4}b^{n/4}\}$

CFG with, for each rule, how many rules it becomes in CNF.

- $S \to ABAB$. CNF: 2 rules.
- $A \to a \cdots a$. $n/4$ $a$’s. CNF: $\log_2(n/4)$ rules.
- $B \to b \cdots b$. $n/4$ $b$’s. CNF: $\log_2(n/4)$ rules.

**Number of Rules:**

$2 + 2 \log_2(n/4) = 2 + 2(\log_2(n) - 2) = 2 \log_2(n) - 2.$
Prob 3: CFG for \( L = \{a^n b^n c^n : n \in \mathbb{N}\} \)

We give \( L \) as a \( \cup \) of set, each of which is reg or CFL.
We give $L$ as a $\bigcup$ of set, each of which is reg or CFL. We first present sets where the $a$’s, $b$’s, $c$’s are out of order.
Prob 3: CFG for \( L = \{a^n b^n c^n : n \in \mathbb{N}\} \)

We give \( L \) as a \( \cup \) of set, each of which is reg or CFL. We first present sets where the \( a \)'s, \( b \)'s, \( c \)'s are out of order.

1. \( \{a, b, c\}^* ba\{a, b, c\}^* \). This is regular.
Prob 3: CFG for \( L = \{a^n b^n c^n : n \in \mathbb{N}\} \)

We give \( L \) as a \( \cup \) of set, each of which is reg or CFL. We first present sets where the \( a \)'s, \( b \)'s, \( c \)'s are out of order.

1. \( \{a, b, c\}^* ba\{a, b, c\}^* \). This is regular.
2. \( \{a, b, c\}^* cb\{a, b, c\}^* \). This is regular.
We give $L$ as a $\cup$ of set, each of which is reg or CFL. We first present sets where the $a$’s, $b$’s, $c$’s are out of order.

1. $\{a, b, c\}^* ba \{a, b, c\}^*$. This is regular.
2. $\{a, b, c\}^* cb \{a, b, c\}^*$. This is regular.
3. $\{a, b, c\}^* ca \{a, b, c\}^*$. This is regular.
Prob 3: CFG for $L = \{a^n b^n c^n : n \in \mathbb{N}\}$

We give $L$ as a $\cup$ of set, each of which is reg or CFL. We first present sets where the $a$’s, $b$’s, $c$’s are out of order.

1. $\{a, b, c\}^* ba\{a, b, c\}^*$. This is regular.
2. $\{a, b, c\}^* cb\{a, b, c\}^*$. This is regular.
3. $\{a, b, c\}^* ca\{a, b, c\}^*$. This is regular.

Next slide is the sets that are of the form $a^* b^* c^*$ but have the numbers-of-symbols wrong.
Prob 3: CFG for $L = \{a^n b^n c^n : n \in \mathbb{N}\}$ (cont)

1. $\{a^m b^n : m > n\} \cdot c^*$.
   
   $c^*$ is reg, hence CFL. CFL’s are closed under concat. we need only give a CFG for
Prob 3: CFG for $L = \{a^n b^n c^n : n \in \mathbb{N}\}$ (cont)

1. $\{a^m b^n : m > n\} \cdot c^*$. 
   $c^*$ is reg, hence CFL. CFL’s are closed under concat. we need only give a CFG for 
   $L_1 = \{a^m b^n : m > n\}$
Prob 3: CFG for \( L = \{a^n b^n c^n : n \in \mathbb{N}\}\) (cont)

1. \( \{a^m b^n : m > n\} \cdot c^* \).  
   \( c^* \) is reg, hence CFL. CFL’s are closed under concat. we need only give a CFG for  
   \( L_1 = \{a^m b^n : m > n\} \)  
   \( S \rightarrow AT \)  
   \( T \rightarrow aTb \mid e \)  
   \( A \rightarrow Aa \mid a \)  
   (The remaining sets are similar.)
1. \( \{a^m b^n : m > n\} \cdot c^* \).
   
   \( c^* \) is reg, hence CFL. CFL’s are closed under concat. we need only give a CFG for
   
   \( L_1 = \{a^m b^n : m > n\} \)
   
   \( S \rightarrow AT \)
   
   \( T \rightarrow aTb \mid e \)
   
   \( A \rightarrow Aa \mid a \)
   
   (The remaining sets are similar.)

2. \( \{a^m b^n c^* : m < n\} \)
Prob 3: CFG for \( L = \{a^n b^n c^n : n \in \mathbb{N}\}\) (cont)

1. \(\{a^m b^n : m > n\} \cdot c^*\).
   
   \(c^*\) is reg, hence CFL. CFL’s are closed under concat. we need only give a CFG for

   \[ L_1 = \{a^m b^n : m > n\} \]

   \[ S \rightarrow AT \]

   \[ T \rightarrow aTb \quad | \quad e \]

   \[ A \rightarrow Aa \quad | \quad a \]

   (The remaining sets are similar.)

2. \(\{a^m b^n c^* : m < n\}\)

3. \(\{a^* b^m c^n : m > n\}\)
Prob 3: CFG for $L = \{a^n b^n c^n : n \in \mathbb{N}\}$ (cont)

1. $\{a^m b^n : m > n\} \cdot c^*$. 
   $c^*$ is reg, hence CFL. CFL’s are closed under concat. we need only give a CFG for 
   $L_1 = \{a^m b^n : m > n\}$
   $S \rightarrow AT$
   $T \rightarrow aTb \mid e$
   $A \rightarrow Aa \mid a$
   (The remaining sets are similar.)

2. $\{a^m b^n c^* : m < n\}$

3. $\{a^* b^m c^n : m > n\}$

4. $\{a^* b^m c^n : m < n\}$
Prob 3: CFG for \( L = \{a^n b^n c^n : n \in \mathbb{N}\} \)

1. \( \{a^m b^n : m > n\} \cdot c^* \).
   
   \( c^* \) is reg, hence CFL. CFL’s are closed under concat. we need only give a CFG for
   
   \( L_1 = \{a^m b^n : m > n\} \)

   \( S \rightarrow AT \)

   \( T \rightarrow aTb \mid e \)

   \( A \rightarrow Aa \mid a \)

   (The remaining sets are similar.)

2. \( \{a^m b^n c^* : m < n\} \)

3. \( \{a^* b^m c^n : m > n\} \)

4. \( \{a^* b^m c^n : m < n\} \)

5. \( \{a^m b^* c^n : m > n\} \)
Prob 3: CFG for \( L = \{a^n b^n c^n : n \in \mathbb{N}\} \) (cont)

1. \( \{a^m b^n : m > n\} \cdot c^* \).
   
   \( c^* \) is reg, hence CFL. CFL’s are closed under concat. we need only give a CFG for
   
   \( L_1 = \{a^m b^n : m > n\} \)

   \[ S \rightarrow AT \]
   
   \[ T \rightarrow aTb \mid e \]
   
   \[ A \rightarrow Aa \mid a \]

   (The remaining sets are similar.)

2. \( \{a^m b^n c^* : m < n\} \)

3. \( \{a^* b^m c^n : m > n\} \)

4. \( \{a^* b^m c^n : m < n\} \)

5. \( \{a^m b^* c^n : m > n\} \)

6. \( \{a^m b^* c^n : m < n\} \)
Prob 4a: DFA for $L = \{ w : |w| = n \land \#a(w) = n/2 \}$
Prob 4a: DFA for \( L = \{ w : |w| = n \land \#_a(w) = n/2 \} \)

DFA keeps track of \( |w| \) and \( \#_a(w) \).
Prob 4a: DFA for $L = \{w : |w| = n \land \#_a(w) = n/2\}$

DFA keeps track of $|w|$ and $\#_a(w)$. 
$Q = \{(i, j) : 1 \leq i \leq n \text{ AND } j \leq \frac{n}{2} \text{ AND } j \leq i\} \cup \{d\}$
Prob 4a: DFA for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

DFA keeps track of $|w|$ and $\#_a(w)$.

$Q = \{(i,j) : 1 \leq i \leq n \land j \leq \frac{n}{2} \land j \leq i\} \cup \{d\}$

State $(i,j)$: $i$ chars seen, $j$ of them are a’s. $d$ is dump.
Prob 4a: DFA for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

DFA keeps track of $|w|$ and $\#_a(w)$.
$Q = \{(i, j) : 1 \leq i \leq n \land j \leq \frac{n}{2} \land j \leq i\} \cup \{d\}$
State $(i, j)$: $i$ chars seen, $j$ of them are $a$'s. $d$ is dump.
We describe $\delta$ on the ordered pairs and then $\delta$ on $d$. 
Prob 4a: DFA for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

DFA keeps track of $|w|$ and $\#_a(w)$.
$Q = \{(i, j) : 1 \leq i \leq n \land j \leq \frac{n}{2} \land j \leq i \} \cup \{d\}$
State $(i, j)$: $i$ chars seen, $j$ of them are $a$’s. $d$ is dump.
We describe $\delta$ on the ordered pairs and then $\delta$ on $d$.
For $1 \leq i \leq n$, $1 \leq j \leq \frac{n}{2}$, and $\sigma \in \{a, b\}$:
Prob 4a: DFA for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

DFA keeps track of $|w|$ and $\#_a(w)$.
$Q = \{ (i, j) : 1 \leq i \leq n \land j \leq \frac{n}{2} \land j \leq i \} \cup \{ d \}$
State $(i, j)$: $i$ chars seen, $j$ of them are $a$’s. $d$ is dump.
We describe $\delta$ on the ordered pairs and then $\delta$ on $d$.
For $1 \leq i \leq n$, $1 \leq j \leq \frac{n}{2}$, and $\sigma \in \{ a, b \}$:

$$
\delta((i, j), \sigma) = \begin{cases} 
(i + 1, j), & \text{if } i \leq n - 1 \text{ and } \sigma = b \\
(i + 1, j + 1), & \text{if } i \leq n - 1 \text{ and } \sigma = a \text{ and } j \leq \frac{n}{2} \\
d, & \text{if } i = n \lor (\sigma = a \land j = \frac{n}{2}) 
\end{cases}
$$

$F = \{ (n, n/2) \}.$ Can we do better? U Vote: Can do better, can’t do better, UNK TO BILL.
Prob 4a: DFA for \( L = \{ w : |w| = n \land \#_a(w) = n/2 \} \)

DFA keeps track of \(|w|\) and \(\#_a(w)\).

\[ Q = \{(i, j) : 1 \leq i \leq n \text{ AND } j \leq \frac{n}{2} \text{ AND } j \leq i\} \cup \{d\} \]

State \((i, j)\): \(i\) chars seen, \(j\) of them are \(a\)'s. \(d\) is dump.

We describe \(\delta\) on the ordered pairs and then \(\delta\) on \(d\).

For \(1 \leq i \leq n\), \(1 \leq j \leq \frac{n}{2}\), and \(\sigma \in \{a, b\}\):

\[
\delta((i, j), \sigma) = \begin{cases} 
(i + 1, j) & \text{, if } i \leq n - 1 \text{ and } \sigma = b \\
(i + 1, j + 1) & \text{, if } i \leq n - 1 \text{ and } \sigma = a \text{ and } j \leq \frac{n}{2} \\
d & \text{, if } i = n \lor (\sigma = a \land j = \frac{n}{2})
\end{cases}
\]

For \(\sigma \in \{a, b\}\), \(\delta(d, \sigma)\) is defined by \(\delta(d, \sigma) = d\).
Prob 4a: DFA for \( L = \{ w : |w| = n \wedge \#_a(w) = n/2 \} \)

DFA keeps track of \(|w|\) and \(\#_a(w)\).

\[ Q = \{(i, j) : 1 \leq i \leq n \text{ AND } j \leq \frac{n}{2} \text{ AND } j \leq i\} \cup \{d\} \]

State \((i, j)\): \(i\) chars seen, \(j\) of them are \(a\)'s. \(d\) is dump.

We describe \(\delta\) on the ordered pairs and then \(\delta\) on \(d\).

For \(1 \leq i \leq n, 1 \leq j \leq \frac{n}{2}\), and \(\sigma \in \{a, b\}\):

\[
\delta((i, j), \sigma) = \begin{cases} 
(i + 1, j) & \text{, if } i \leq n - 1 \text{ and } \sigma = b \\
(i + 1, j + 1) & \text{, if } i \leq n - 1 \text{ and } \sigma = a \text{ and } j \leq \frac{n}{2} \\
d & \text{, if } i = n \lor (\sigma = a \land j = \frac{n}{2})
\end{cases}
\]

For \(\sigma \in \{a, b\}\), \(\delta(d, \sigma)\) is defined by \(\delta(d, \sigma) = d\).

\(F = \{(n, n/2)\}\).
Prob 4a: DFA for $L = \{w : |w| = n \land \#_a(w) = n/2\}$

DFA keeps track of $|w|$ and $\#_a(w)$.
$Q = \{(i, j) : 1 \leq i \leq n \land j \leq \frac{n}{2} \land j \leq i\} \cup \{d\}$
State $(i, j)$: $i$ chars seen, $j$ of them are $a$’s. $d$ is dump.
We describe $\delta$ on the ordered pairs and then $\delta$ on $d$.
For $1 \leq i \leq n$, $1 \leq j \leq \frac{n}{2}$, and $\sigma \in \{a, b\}$:

$$
\delta((i, j), \sigma) = \begin{cases} 
(i + 1, j) & \text{, if } i \leq n - 1 \land \sigma = b \\
(i + 1, j + 1) & \text{, if } i \leq n - 1 \land \sigma = a \land j \leq \frac{n}{2} \\
d & \text{, if } i = n \lor (\sigma = a \land j = \frac{n}{2})
\end{cases}
$$

For $\sigma \in \{a, b\}$, $\delta(d, \sigma)$ is defined by $\delta(d, \sigma) = d$.
$F = \{(n, n/2)\}$. The number of states is $O(n^2)$. Can we do better? U
Prob 4a: DFA for \( L = \{ w : |w| = n \land \#_a(w) = n/2 \} \)

DFA keeps track of \(|w|\) and \(\#_a(w)\).

\[ Q = \{(i, j) : 1 \leq i \leq n \land j \leq \frac{n}{2} \land j \leq i \} \cup \{d\} \]

State \((i, j)\): \(i\) chars seen, \(j\) of them are \(a\)’s. \(d\) is dump.

We describe \(\delta\) on the ordered pairs and then \(\delta\) on \(d\).

For \(1 \leq i \leq n, 1 \leq j \leq \frac{n}{2}\), and \(\sigma \in \{a, b\}\):

\[
\delta((i, j), \sigma) = \begin{cases} 
(i + 1, j) & \text{, if } i \leq n - 1 \text{ and } \sigma = b \\
(i + 1, j + 1) & \text{, if } i \leq n - 1 \text{ and } \sigma = a \land j \leq \frac{n}{2} \\
d & \text{, if } i = n \lor (\sigma = a \land j = \frac{n}{2}) 
\end{cases}
\]

For \(\sigma \in \{a, b\}\), \(\delta(d, \sigma)\) is defined by \(\delta(d, \sigma) = d\).

\[ F = \{(n, n/2)\} \]. The number of states is \(O(n^2)\). Can we do better? U

Vote: Can do better, can’t do better, UNK TO BILL.
Prob 4b: Rgx for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$
Prob 4b: Rgx for $L = \{w: |w| = n \land \#_a(w) = n/2\}$

Note that there are $|L| = \binom{n}{n/2} = \Theta(\frac{2^n}{\sqrt{n}})$. 
Prob 4b: Rgx for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

Note that there are $|L| = \binom{n}{n/2} = \Theta\left(\frac{2^n}{\sqrt{n}}\right)$. Let $N = \binom{n}{n/2}$. 
Prob 4b: Rgx for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

Note that there are $|L| = \binom{n}{n/2} = \Theta(\frac{2^n}{\sqrt{n}})$. Let $N = \binom{n}{n/2}$.

Let all the strings in $L$ be $w_1, w_2, \ldots, w_N$. 
Prob 4b: Rgx for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

Note that there are $|L| = \binom{n}{n/2} = \Theta\left(\frac{2^n}{\sqrt{n}}\right)$. Let $N = \binom{n}{n/2}$. Let all the strings in $L$ be $w_1, w_2, \ldots, w_N$. Regex for $L$:

$$\{ w_1 \} \cup \{ w_2 \} \cup \cdots \cup \{ w_N \}.$$
Prob 4b: Rgx for \( L = \{w : |w| = n \land \#_a(w) = n/2\} \)

Note that there are \( |L| = \binom{n}{n/2} = \Theta\left(\frac{2^n}{\sqrt{n}}\right) \). Let \( N = \binom{n}{n/2} \).

Let all the strings in \( L \) be \( w_1, w_2, \ldots, w_N \). Regex for \( L \):

\[
\{w_1\} \cup \{w_2\} \cup \cdots \cup \{w_N\}.
\]

Length is \( Nn = \binom{n}{n/2}n = O(\sqrt{n}2^n) \).
Prob 4b: Rgx for $L = \{w : |w| = n \land \#_a(w) = n/2\}$

Note that there are $|L| = \binom{n}{n/2} = \Theta\left(\frac{2^n}{\sqrt{n}}\right)$. Let $N = \binom{n}{n/2}$.

Let all the strings in $L$ be $w_1, w_2, \ldots, w_N$. Regex for $L$:

$$\{w_1\} \cup \{w_2\} \cup \cdots \cup \{w_N\}.$$

Length is $Nn = \binom{n}{n/2}n = O(\sqrt{n}2^n)$.

Vote
Note that there are $|L| = \binom{n}{n/2} = \Theta(\frac{2^n}{\sqrt{n}})$. Let $N = \binom{n}{n/2}$.

Let all the strings in $L$ be $w_1, w_2, \ldots, w_N$. Regex for $L$:

$$\{w_1\} \cup \{w_2\} \cup \cdots \cup \{w_N\}.$$

Length is $Nn = \binom{n}{n/2}n = O(\sqrt{n}2^n)$.

**Vote**

There is a poly-sized regex and this is known.
Prob 4b: Rgx for $L = \{w : |w| = n \land \#_a(w) = n/2\}$

Note that there are $|L| = \binom{n}{n/2} = \Theta(\frac{2^n}{\sqrt{n}})$. Let $N = \binom{n}{n/2}$.

Let all the strings in $L$ be $w_1, w_2, \ldots, w_N$. Regex for $L$:

$$\{w_1\} \cup \{w_2\} \cup \cdots \cup \{w_N\}.$$ 

Length is $Nn = \binom{n}{n/2} n = O(\sqrt{n}2^n)$.

**Vote**

There is a poly-sized regex and this is known.
There is not a poly-sized regex and this is known.
Prob 4b: Rgx for \( L = \{ w : |w| = n \land \#_a(w) = n/2 \} \)

Note that there are \(|L| = \binom{n}{n/2} = \Theta\left(\frac{2^n}{\sqrt{n}}\right)\). Let \( N = \binom{n}{n/2} \).

Let all the strings in \( L \) be \( w_1, w_2, \ldots, w_N \). Regex for \( L \):

\[
\{ w_1 \} \cup \{ w_2 \} \cup \cdots \cup \{ w_N \}.
\]

Length is \( Nn = \binom{n}{n/2} n = O(\sqrt{n}2^n) \).

**Vote**

There is a poly-sized regex and this is known.

There is not a poly-sized regex and this is known.

Poly-sized regex or not is **UNK TO BILL**
Prob 4b: Rgx for \( L = \{ w : |w| = n \land \#_a(w) = n/2 \} \)

Note that there are \( |L| = \binom{n}{n/2} = \Theta\left( \frac{2^n}{\sqrt{n}} \right) \). Let \( N = \binom{n}{n/2} \).

Let all the strings in \( L \) be \( w_1, w_2, \ldots, w_N \). Regex for \( L \):

\[
\{ w_1 \} \cup \{ w_2 \} \cup \cdots \cup \{ w_N \}.
\]

Length is \( Nn = \binom{n}{n/2} n = O(\sqrt{n}2^n) \).

**Vote**

There is a poly-sized regex and this is known.
There is not a poly-sized regex and this is known.
Poly-sized regex or not is **UNK TO BILL**
Answer on the next slides.
There is No Poly Sized Regex for $L$.
There is No Poly Sized Regex for $L$

Alphabet is $\{a, b\}$. 

1. Ellul-Kravwetu-Shallit-Wang, in 2005, showed that $L_{n,k}$ has a regex of size $O(n(\log n)^k)$.

Paper is here: https://cs.uwaterloo.ca/~shallit/Papers/re3.pdf

2. If $k = n^2$ this is $2^{\Omega(n)}$. If worked out then probably better than what we got, but not poly.

3. Mousavi, in 2017, showed that any regex for $L_{n,k}$ has length at least $\Omega(n(\log n)^k)$.

Paper is here: https://arxiv.org/abs/1712.00811

4. If $k = n^2$ this is $2^{\Omega(n)}$. So there is no polysized Regex for $L_{n,n/2}$. 
There is No Poly Sized Regex for \( L \)

Alphabet is \( \{a, b\} \).

**Definition** \( L_{n,k} = \{w : |w| = n \wedge \#_a(w) = k\} \).
There is No Poly Sized Regex for $L$

Alphabet is $\{a, b\}$.

**Definition** $L_{n,k} = \{w : |w| = n \land \#_a(w) = k\}$.

1. Ellul-Kravwetu-Shallit-Wang, in 2005, showed that $L_{n,k}$ has a regex of size $O(n(\log n)^k)$.

2. If $k = n^2$ this is $2^{O(n)}$. If worked out then probably better than what we got, but not poly.

3. Mousavi, in 2017, showed that any regex for $L_{n,k}$ has length at least $\Omega(n(\log n)^k)$.

4. If $k = n^2$ this is $2^{\Omega(n)}$.

So there is no polysized Regex for $L_{n,n/2}$. 

Paper is here: https://cs.uwaterloo.ca/~shallit/Papers/re3.pdf

Paper is here: https://arxiv.org/abs/1712.00811
There is No Poly Sized Regex for $L$

Alphabet is $\{a, b\}$.

**Definition** $L_{n,k} = \{ w : |w| = n \land \#_a(w) = k \}$.

1. Ellul-Kravvetu-Shallit-Wang, in 2005, showed that $L_{n,k}$ has a regex of size $O(n(\log n)^k)$.
   
   Paper is here: [https://cs.uwaterloo.ca/~shallit/Papers/re3.pdf](https://cs.uwaterloo.ca/~shallit/Papers/re3.pdf)

2. If $k = n^2$ this is $2^{O(n)}$. If worked out then probably better than what we got, but not poly.

3. Mousavi, in 2017, showed that any regex for $L_{n,k}$ has length at least $\Omega(n(\log n)^k)$.
   

4. If $k = n^2$ this is $2^{\Omega(n)}$.

So there is no polysized Regex for $L_{n,n/2}$. 

There is No Poly Sized Regex for $L$

Alphabet is $\{a, b\}$.

**Definition** $L_{n,k} = \{w : |w| = n \land \#_a(w) = k\}$.

1. Ellul-Kravwetu-Shallit-Wang, in 2005, showed that $L_{n,k}$ has a regex of size $O(n(\log n)^k)$. Paper is here: [https://cs.uwaterloo.ca/~shallit/Papers/re3.pdf](https://cs.uwaterloo.ca/~shallit/Papers/re3.pdf)

2. If $k = \frac{n}{2}$ this is $2^{O(n)}$. If worked out then probably better than what we got, but not poly.
There is No Poly Sized Regex for $L$

Alphabet is $\{a, b\}$.

**Definition** $L_{n,k} = \{ w : |w| = n \land \#_a(w) = k \}$.

1. Ellul-Kravwetu-Shallit-Wang, in 2005, showed that $L_{n,k}$ has a regex of size $O(n(\log n)^k)$.

   Paper is here: https://cs.uwaterloo.ca/~shallit/Papers/re3.pdf

2. If $k = \frac{n}{2}$ this is $2^{O(n)}$. If worked out then probably better than what we got, but not poly.

3. Mousavi, in 2017, showed that any regex for $L_{n,k}$ has length at least $\Omega(n(\log n)^k)$.
There is No Poly Sized Regex for $L$

Alphabet is $\{a, b\}$.

**Definition** $L_{n,k} = \{ w : |w| = n \land \# a(w) = k \}$.

1. Ellul-Kravwetu-Shallit-Wang, in 2005, showed that $L_{n,k}$ has a regex of size $O(n(\log n)^k)$.
   Paper is here: https://cs.uwaterloo.ca/~shallit/Papers/re3.pdf

2. If $k = \frac{n}{2}$ this is $2^{O(n)}$. If worked out then probably better than what we got, but not poly.

3. Mousavi, in 2017, showed that any regex for $L_{n,k}$ has length at least $\Omega(n(\log n)^k)$.
   Paper is here: https://arxiv.org/abs/1712.00811
There is No Poly Sized Regex for $L$

Alphabet is $\{a, b\}$.

**Definition** $L_{n,k} = \{w : |w| = n \land \#_a(w) = k\}$.

1. Ellul-Kravvetu-Shallit-Wang, in 2005, showed that $L_{n,k}$ has a regex of size $O(n(\log n)^k)$. Paper is here: https://cs.uwaterloo.ca/~shallit/Papers/re3.pdf
2. If $k = \frac{n}{2}$ this is $2^{O(n)}$. If worked out then probably better than what we got, but not poly.
3. Mousavi, in 2017, showed that any regex for $L_{n,k}$ has length at least $\Omega(n(\log n)^k)$. Paper is here: https://arxiv.org/abs/1712.00811
4. If $k = \frac{n}{2}$ this is $2^{\Omega(n)}$. 
There is No Poly Sized Regex for $L$

Alphabet is $\{a, b\}$.

**Definition** $L_{n,k} = \{w : |w| = n \land \#_a(w) = k\}$.

1. Ellul-Kravvetu-Shallit-Wang, in 2005, showed that $L_{n,k}$ has a regex of size $O(n(\log n)^k)$.
   Paper is here: https://cs.uwaterloo.ca/~shallit/Papers/re3.pdf

2. If $k = \frac{n}{2}$ this is $2^{O(n)}$. If worked out then probably better than what we got, but not poly.

3. Mousavi, in 2017, showed that any regex for $L_{n,k}$ has length at least $\Omega(n(\log n)^k)$.
   Paper is here: https://arxiv.org/abs/1712.00811

4. If $k = \frac{n}{2}$ this is $2^{\Omega(n)}$.

So there is no polysized Regex for $L_{n,n/2}$.
Prob 4c: CFG for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$
Prob 4c: CFG for \( L = \{ w : |w| = n \land \#_a(w) = n/2 \} \)

Note that there are \( |L| = \binom{n}{n/2} = \Theta\left(\frac{2^n}{\sqrt{n}}\right) \).
Prob 4c: CFG for \( L = \{ w : |w| = n \land \#_a(w) = n/2 \} \)

Note that there are \( |L| = \binom{n}{n/2} = \Theta\left(\frac{2^n}{\sqrt{n}}\right) \). Let \( N = \binom{n}{n/2} \).
Prob 4c: CFG for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

Note that there are $|L| = \binom{n}{n/2} = \Theta(\frac{2^n}{\sqrt{n}})$. Let $N = \binom{n}{n/2}$.

Let all the strings in $L$ be $w_1, w_2, \ldots, w_N$. 

Prob 4c: CFG for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

Note that there are $|L| = \binom{n}{n/2} = \Theta(\frac{2^n}{\sqrt{n}})$. Let $N = \binom{n}{n/2}$.

Let all the strings in $L$ be $w_1, w_2, \ldots, w_N$.

$(\forall i) G_i$ is CFG for $\{w_i\}$ of size $O(n)$. $S_i$ is start sym of $G_i$. 

Vote: Poly-known, not-Poly-known, UNK TO BILL.
Prob 4c: CFG for $L = \{w : |w| = n \land \#_a(w) = n/2\}$

Note that there are $|L| = \binom{n}{n/2} = \Theta\left(\frac{2^n}{\sqrt{n}}\right)$. Let $N = \binom{n}{n/2}$.

Let all the strings in $L$ be $w_1, w_2, \ldots, w_N$.

$(\forall i) G_i$ is CFG for $\{w_i\}$ of size $O(n)$. $S_i$ is start sym of $G_i$.

The CFL:
Prob 4c: CFG for $L = \{w : |w| = n \land \#_a(w) = n/2\}$

Note that there are $|L| = \binom{n}{n/2} = \Theta(\frac{2^n}{\sqrt{n}})$. Let $N = \binom{n}{n/2}$.

Let all the strings in $L$ be $w_1, w_2, \ldots, w_N$.

$(\forall i) \ G_i$ is CFG for $\{w_i\}$ of size $O(n)$. $S_i$ is start sym of $G_i$.

The CFL:

1. Start State is $S$. For all $i$ add $S \to S_i$. 
Prob 4c: CFG for \( L = \{ w : |w| = n \land \#_{a}(w) = n/2 \} \)

Note that there are \(|L| = \binom{n}{n/2} = \Theta(\frac{2^n}{\sqrt{n}})\). Let \( N = \binom{n}{n/2} \).

Let all the strings in \( L \) be \( w_1, w_2, \ldots, w_N \).

(\( \forall i \)) \( G_i \) is CFG for \( \{ w_i \} \) of size \( O(n) \). \( S_i \) is start sym of \( G_i \).

The CFL:

1. Start State is \( S \). For all \( i \) add \( S \rightarrow S_i \).
2. Add all of the rules of all of the \( G_i \)'s.
Prob 4c: CFG for $L = \{w : |w| = n \land \#_a(w) = n/2\}$

Note that there are $|L| = \binom{n}{n/2} = \Theta\left(\frac{2^n}{\sqrt{n}}\right)$. Let $N = \binom{n}{n/2}$.

Let all the strings in $L$ be $w_1, w_2, \ldots, w_N$.

$(\forall i)$ $G_i$ is CFG for $\{w_i\}$ of size $O(n)$. $S_i$ is start sym of $G_i$.

The CFL:

1. Start State is $S$. For all $i$ add $S \rightarrow S_i$.
2. Add all of the rules of all of the $G_i$’s.

$N$ $G_i$’s. Each has $O(n)$ rules. $G$ has
Prob 4c: CFG for $L = \{ w : |w| = n \wedge \#_a(w) = n/2 \}$

Note that there are $|L| = \binom{n}{n/2} = \Theta\left(\frac{2^n}{\sqrt{n}}\right)$. Let $N = \binom{n}{n/2}$.

Let all the strings in $L$ be $w_1, w_2, \ldots, w_N$.

$(\forall i) \ G_i$ is CFG for $\{w_i\}$ of size $O(n)$. $S_i$ is start sym of $G_i$.

The CFL:

1. Start State is $S$. For all $i$ add $S \rightarrow S_i$.
2. Add all of the rules of all of the $G_i$’s.

$N \ G_i$’s. Each has $O(n)$ rules. $G$ has

$$O(Nn) = O\left(\left(\binom{n}{n/2}\right)n\right) = O(\sqrt{n}2^n) \text{rules.}$$
Prob 4c: CFG for $L = \{ w : \# w = n / 2 \}$

Note that there are $|L| = \binom{n}{n/2} = \Theta\left(\frac{2^n}{\sqrt{n}}\right)$. Let $N = \binom{n}{n/2}$.

Let all the strings in $L$ be $w_1, w_2, \ldots, w_N$.

$(\forall i)$ $G_i$ is CFG for $\{w_i\}$ of size $O(n)$. $S_i$ is start sym of $G_i$.

The CFL:

1. Start State is $S$. For all $i$ add $S \to S_i$.
2. Add all of the rules of all of the $G_i$’s.

$N$ $G_i$’s. Each has $O(n)$ rules. $G$ has

$$O(Nn) = O\left(\left(\frac{n}{n/2}\right)^n\right) = O(\sqrt{n}2^n)$$

rules.

**Vote**: Poly-known, not-Poly-known, **UNK TO BILL**.
Prob 4c: CFG for $L = \{w : |w| = n \land \#_a(w) = n/2\}$

Note that there are $|L| = \binom{n}{n/2} = \Theta\left(\frac{2^n}{\sqrt{n}}\right)$. Let $N = \binom{n}{n/2}$.

Let all the strings in $L$ be $w_1, w_2, \ldots, w_N$.

$(\forall i)$ $G_i$ is CFG for $\{w_i\}$ of size $O(n)$. $S_i$ is start sym of $G_i$.

The CFL:
1. Start State is $S$. For all $i$ add $S \rightarrow S_i$.
2. Add all of the rules of all of the $G_i$’s.

$N$ $G_i$’s. Each has $O(n)$ rules. $G$ has

$$O(Nn) = O\left(\left(\binom{n}{n/2}\right)n\right) = O(\sqrt{n}2^n)$$

rules.

Vote: Poly-known, not-Poly-known, \textbf{UNK TO BILL}.

Answer on next slide.
Poly Size CFG for $L = \{w : |w| = n \wedge \#_a(w) = n/2\}$

**Key** Small CFG for $L_{i,j} = \{w : |w| = i \wedge \#_a(w) = j\}$
Poly Size CFG for $L = \{w : |w| = n \land \#_a(w) = n/2\}$

Key Small CFG for $L_{i,j} = \{w : |w| = i \land \#_a(w) = j\}$
The start symbol for the CFG for $L_{i,j}$ will be $S_{i,j}$.
Poly Size CFG for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

Key Small CFG for $L_{i,j} = \{ w : |w| = i \land \#_a(w) = j \}$
The start symbol for the CFG for $L_{i,j}$ will be $S_{i,j}$.

1. $L_{i,0} = \{ b^i \}$. 
Poly Size CFG for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

**Key** Small CFG for $L_{i,j} = \{ w : |w| = i \land \#_a(w) = j \}$

The start symbol for the CFG for $L_{i,j}$ will be $S_{i,j}$.

1. $L_{i,0} = \{ b^i \}$. CFG-CNF with $O(\log i)$ rules.
Poly Size CFG for \( L = \{ w : |w| = n \land \#_a(w) = n/2 \} \)

**Key** Small CFG for \( L_{i,j} = \{ w : |w| = i \land \#_a(w) = j \} \)

The start symbol for the CFG for \( L_{i,j} \) will be \( S_{i,j} \).

1. \( L_{i,0} = \{ b^i \} \). CFG-CNF with \( O(\log i) \) rules.
2. \( L_{1,1} = \{ a \} \).

"I am sure you can all go home and prove that by induction."

The Grammar is of size \( O(n^2) \).

**Upshot** There is a CFG for \( L_n, n/2 \) of size \( O(n^2) \).
Poly Size CFG for \( L = \{ w : |w| = n \land \#_a(w) = n/2 \} \)

**Key** Small CFG for \( L_{i,j} = \{ w : |w| = i \land \#_a(w) = j \} \)

The start symbol for the CFG for \( L_{i,j} \) will be \( S_{i,j} \).

1. \( L_{i,0} = \{ b^i \} \). CFG-CNF with \( O(\log i) \) rules.
2. \( L_{1,1} = \{ a \} \). CFG-CNF with \( O(1) \) rules.

"I am sure you can all go home and prove that by induction."

The Grammar is of size \( O(n^2) \).

**Upshot** There is a CFG for \( L_{n,n/2} \) of size \( O(n^2) \).
Poly Size CFG for $L = \{ w : |w| = n \wedge \#_a(w) = n/2 \}$

**Key** Small CFG for $L_{i,j} = \{ w : |w| = i \wedge \#_a(w) = j \}$

The start symbol for the CFG for $L_{i,j}$ will be $S_{i,j}$.

1. $L_{i,0} = \{ b^i \}$. CFG-CNF with $O(\log i)$ rules.
2. $L_{1,1} = \{ a \}$. CFG-CNF with $O(1)$ rules.
3. $L_{1,0} = \{ b \}$. 

"I am sure you can all go home and prove that by induction."

The Grammar is of size $O(n^2)$. 

Upshot

There is a CFG for $L_{n,n}$ of size $O(n^2)$. 
Poly Size CFG for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

Key Small CFG for $L_{i,j} = \{ w : |w| = i \land \#_a(w) = j \}$

The start symbol for the CFG for $L_{i,j}$ will be $S_{i,j}$.

1. $L_{i,0} = \{ b^i \}$. CFG-CNF with $O(\log i)$ rules.
2. $L_{1,1} = \{ a \}$. CFG-CNF with $O(1)$ rules.
3. $L_{1,0} = \{ b \}$. CFG-CNF with $O(1)$ rules.

"I am sure you can all go home and prove that by induction."

The Grammar is of size $O(n^2)$. 

Upshot There is a CFG for $L_n, n/2$ of size $O(n^2)$. 

Poly Size CFG for \( L = \{ w : |w| = n \land \#_a(w) = n/2 \} \)

**Key** Small CFG for \( L_{i,j} = \{ w : |w| = i \land \#_a(w) = j \} \)

The start symbol for the CFG for \( L_{i,j} \) will be \( S_{i,j} \).

1. \( L_{i,0} = \{ b^i \} \). CFG-CNF with \( O(\log i) \) rules.
2. \( L_{1,1} = \{ a \} \). CFG-CNF with \( O(1) \) rules.
3. \( L_{1,0} = \{ b \} \). CFG-CNF with \( O(1) \) rules.
4. For \( 2 \leq i \leq n, 1 \leq j \leq i \) add the rules:
Poly Size CFG for \( L = \{ w : |w| = n \land \#_a(w) = n/2 \} \)

**Key** Small CFG for \( L_{i,j} = \{ w : |w| = i \land \#_a(w) = j \} \)

The start symbol for the CFG for \( L_{i,j} \) will be \( S_{i,j} \).

1. \( L_{i,0} = \{ b^i \} \). CFG-CNF with \( O(\log i) \) rules.
2. \( L_{1,1} = \{ a \} \). CFG-CNF with \( O(1) \) rules.
3. \( L_{1,0} = \{ b \} \). CFG-CNF with \( O(1) \) rules.
4. For \( 2 \leq i \leq n, 1 \leq j \leq i \) add the rules:
   \[
   S_{i,j} \rightarrow aS_{i−1,j−1}
   \]
Poly Size CFG for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

**Key** Small CFG for $L_{i,j} = \{ w : |w| = i \land \#_a(w) = j \}$

The start symbol for the CFG for $L_{i,j}$ will be $S_{i,j}$.

1. $L_{i,0} = \{b^i\}$. CFG-CNF with $O(\log i)$ rules.
2. $L_{1,1} = \{a\}$. CFG-CNF with $O(1)$ rules.
3. $L_{1,0} = \{b\}$. CFG-CNF with $O(1)$ rules.
4. For $2 \leq i \leq n$, $1 \leq j \leq i$ add the rules:
   
   $S_{i,j} \rightarrow aS_{i-1,j-1}$
   $S_{i,j} \rightarrow bS_{i-1,j}$

"I am sure you can all go home and prove that by induction."

The Grammar is of size $O(n^2)$. 

**Upshot** There is a CFG for $L_{n,n}$ of size $O(n^2)$. 
Poly Size CFG for $L = \{w : |w| = n \land \#_a(w) = n/2\}$

**Key** Small CFG for $L_{i,j} = \{w : |w| = i \land \#_a(w) = j\}$

The start symbol for the CFG for $L_{i,j}$ will be $S_{i,j}$.

1. $L_{i,0} = \{b^i\}$. CFG-CNF with $O(\log i)$ rules.
2. $L_{1,1} = \{a\}$. CFG-CNF with $O(1)$ rules.
3. $L_{1,0} = \{b\}$. CFG-CNF with $O(1)$ rules.
4. For $2 \leq i \leq n$, $1 \leq j \leq i$ add the rules:
   - $S_{i,j} \rightarrow aS_{i-1,j-1}$
   - $S_{i,j} \rightarrow bS_{i-1,j}$

If $S_{i,j}$ is Start then get $L_{i,j}$.

"I am sure you can all go home and prove that by induction."

The Grammar is of size $O(n^2)$. 

**Upshot** There is a CFG for $L_{n,n/2}$ of size $O(n^2)$. 

- $L_{i,0} = \{b^i\}$. CFG-CNF with $O(\log i)$ rules.
- $L_{1,1} = \{a\}$. CFG-CNF with $O(1)$ rules.
- $L_{1,0} = \{b\}$. CFG-CNF with $O(1)$ rules.
- For $2 \leq i \leq n$, $1 \leq j \leq i$ add the rules:
  - $S_{i,j} \rightarrow aS_{i-1,j-1}$
  - $S_{i,j} \rightarrow bS_{i-1,j}$

If $S_{i,j}$ is Start then get $L_{i,j}$. 

"I am sure you can all go home and prove that by induction."

The Grammar is of size $O(n^2)$. 

**Upshot** There is a CFG for $L_{n,n/2}$ of size $O(n^2)$.
Key Small CFG for $L_{i,j} = \{w : |w| = i \land \#_a(w) = j\}$

The start symbol for the CFG for $L_{i,j}$ will be $S_{i,j}$.

1. $L_{i,0} = \{b^i\}$. CFG-CNF with $O(\log i)$ rules.
2. $L_{1,1} = \{a\}$. CFG-CNF with $O(1)$ rules.
3. $L_{1,0} = \{b\}$. CFG-CNF with $O(1)$ rules.
4. For $2 \leq i \leq n$, $1 \leq j \leq i$ add the rules:
   
   $S_{i,j} \rightarrow aS_{i-1,j-1}$
   
   $S_{i,j} \rightarrow bS_{i-1,j}$

If $S_{i,j}$ is Start then get $L_{i,j}$.

“I am sure you can all go home and prove that by induction.”
Poly Size CFG for \( L = \{ w : |w| = n \land \#_a(w) = n/2 \} \)

**Key** Small CFG for \( L_{i,j} = \{ w : |w| = i \land \#_a(w) = j \} \)

The start symbol for the CFG for \( L_{i,j} \) will be \( S_{i,j} \).

1. \( L_{i,0} = \{ b^i \} \). CFG-CNF with \( O(\log i) \) rules.
2. \( L_{1,1} = \{ a \} \). CFG-CNF with \( O(1) \) rules.
3. \( L_{1,0} = \{ b \} \). CFG-CNF with \( O(1) \) rules.
4. For \( 2 \leq i \leq n, 1 \leq j \leq i \) add the rules:
   \[
   S_{i,j} \rightarrow aS_{i-1,j-1} \\
   S_{i,j} \rightarrow bS_{i-1,j}
   \]

If \( S_{i,j} \) is Start then get \( L_{i,j} \).

“\( I \) am sure you can all go home and prove that by induction.”

The Grammar is of size \( O(n^2) \).
Poly Size CFG for $L = \{ w : |w| = n \land \#_a(w) = n/2 \}$

**Key** Small CFG for $L_{i,j} = \{ w : |w| = i \land \#_a(w) = j \}$

The start symbol for the CFG for $L_{i,j}$ will be $S_{i,j}$.

1. $L_{i,0} = \{ b^i \}$. CFG-CNF with $O(\log i)$ rules.
2. $L_{1,1} = \{ a \}$. CFG-CNF with $O(1)$ rules.
3. $L_{1,0} = \{ b \}$. CFG-CNF with $O(1)$ rules.
4. For $2 \leq i \leq n$, $1 \leq j \leq i$ add the rules:
   - $S_{i,j} \rightarrow aS_{i-1,j-1}$
   - $S_{i,j} \rightarrow bS_{i-1,j}$

If $S_{i,j}$ is Start then get $L_{i,j}$.

“I am sure you can all go home and prove that by induction.”

The Grammar is of size $O(n^2)$.

**Upshot** There is a CFG for $L_{n,n/2}$ of size $O(n^2)$. 
Bill’s Usual Mantra: RESPECT for Lower Bounds

(I obviously made up these slides before class began, so what I say here might not be true.)
(I obviously made up these slides before class began, so what I say here might not be true.)
Most of you voted as follows:

1. \( L_n, n/2 \) did NOT have a poly sized CFG-CNF.
2. UNK TO BILL
   To show that \( X \) does not exist you need to show that there is no clever idea and there is no hard math that will show that \( X \) does exist.
   In this case someone clever did come along with a solution.
   When was \( L_n, k \) proven to have a small grammar, and by who?
   Bill Gasarch while preparing hw05 in February 2024.
   No New Ideas: It used Dynamic Programming.
Bill’s Usual Mantra: RESPECT for Lower Bounds

(I obviously made up these slides before class began, so what I say here might not be true.)

Most of you voted as follows:

1. $L_{n,n/2}$ did NOT have a poly sized CFG-CNF.
Bill’s Usual Mantra: RESPECT for Lower Bounds

(I obviously made up these slides before class began, so what I say here might not be true.)

Most of you voted as follows:

1. $L_{n,n/2}$ did NOT have a poly sized CFG-CNF.
2. UNK TO BILL
Bill’s Usual Mantra: RESPECT for Lower Bounds

(I obviously made up these slides before class began, so what I say here might not be true.)
Most of you voted as follows:

1. $L_{n,n/2}$ did NOT have a poly sized CFG-CNF.
2. UNK TO BILL

To show that X does not exist you need to show that there is no clever idea and there is no hard math that will show that $X$ does exist.
(I obviously made up these slides before class began, so what I say here might not be true.)

Most of you voted as follows:

1. $L_{n,n/2}$ did NOT have a poly sized CFG-CN.F.
2. **UNK TO BILL**

To show that $X$ **does not exist** you need to show that there **is no clever idea** and there **is no hard math** that will show that $X$ does exist.

In this case someone clever did come along with a solution.
Bill’s Usual Mantra: RESPECT for Lower Bounds

(I obviously made up these slides before class began, so what I say here might not be true.)

Most of you voted as follows:

1. $L_{n,n/2}$ did NOT have a poly sized CFG-CNF.
2. **UNK TO BILL**

To show that $X$ **does not exist** you need to show that **there is no clever idea** and **there is no hard math** that will show that $X$ does exist.

In this case someone clever did come along with a solution. When was $L_{n,k}$ proven to have a small grammar, and by who?
Bill’s Usual Mantra: RESPECT for Lower Bounds

(I obviously made up these slides before class began, so what I say here might not be true.)
Most of you voted as follows:

1. $L_{n,n/2}$ did NOT have a poly sized CFG-CNF.
2. **UNK TO BILL**

To show that $X$ **does not exist** you need to show that **there is no clever idea** and **there is no hard math** that will show that $X$ does exist.

In this case someone clever did come along with a solution.
When was $L_{n,k}$ proven to have a small grammar, and by who? Bill Gasarch while preparing hw05 in February 2024.
(I obviously made up these slides before class began, so what I say here might not be true.)

Most of you voted as follows:

1. $L_{n,n/2}$ did NOT have a poly sized CFG-CNF.
2. UNK TO BILL

To show that $X$ does not exist you need to show that there is no clever idea and there is no hard math that will show that $X$ does exist.

In this case someone clever did come along with a solution. When was $L_{n,k}$ proven to have a small grammar, and by who? Bill Gasarch while preparing hw05 in February 2024.

No New Ideas: It used Dynamic Programming.