Switching Lemma: Let $P$ be a $k$-CNF formula. Pick a random restriction $p$ leaving $p_n$ variables unset. Then there exists $c > 0$ s.t.

$$\Pr \left[ \text{CDT}[P, 1_p] \geq D \right] \leq (c \cdot p \cdot k)^D$$

Proof idea: We'll code each $p$ with $\text{CDT}[-p, 1_p] \geq D$ by a restriction, $p_{code}$, leaving $p_n - D$ variables unset and a 'hint' - a short message, of length $D \cdot [\log k + 2]$ bits.

There are relatively few $(p_{code}, \text{hint})$ pairs possible, so few $p$'s have $\text{CDT}[P, 1_p] \geq D$.

Example:

```
\begin{array}{c}
\text{true, so prune } c_1 \\
\text{set to one}
\end{array}
```

Diagram of the CNF formula with variables and clauses represented.
We want to code 'long paths' in the decision tree.

Define $D$ to be the

$D'$ to be # of distinct variables queried.

$P_{code} = P \circ \sigma_1 \circ \ldots \circ \sigma_D$,

$P_{code} : P_{n-D}$ variables, unset

|Hint| : $D(\log k + 2)$ bits

Decoding algorithm: look at $P_{1P_{code}}$.

Let $c_i$ be the first clause that is falsified.

Look @ hint to tell us which variables in $c_i$ were unset.

Hint tells us how these values were set in $\sigma$.

We set them to their values in $\sigma_i$, and repeat

until $D$ variables unset have been recovered.

What if our hint is random garbage? What's the probability of recovering $P$?

Imagine $P$ is random, and random 'is $P$ with some

$\times$ replaced by random bits.

$n - pn + D$ bits, $D$ of which are imposters.

All we do is pick randomly: we get what we want w.p.

$$\frac{\frac{\frac{\frac{D}{n-pn+D} \ldots \frac{1}{n-pn+1}}{D-1}}{n-pn+D-1}}{n-pn} \leq \left(\frac{\frac{D}{n-pn}}{D-1}\right)^D$$
The chance that \( \text{random} = \text{code} \) is \( \left( 2^D \binom{p^n}{D} \right)^{-1} \)

Probability that \( \text{hint} \text{ random} = \text{hint} = \left( \frac{1}{4k} \right)^D \)

\[
\left( \frac{D}{n-p} \right)^D \geq \Pr \left[ \text{Decide} \left[ \text{random}, \text{hint random} \right] = \rho \right]
\]

\[
= \frac{\Pr \left[ \text{Pr} \left[ P \mid \rho \right] = D \right]}{\Pr \left[ \text{Pr} \left[ \text{hint random} = \text{hint} \right] \right]} \Pr \left[ \text{Pr} \left[ \text{random} = \text{case} \right] \right]
\]

We also have:

\[
\Pr \left[ \text{decide} \left[ P \mid \rho \right] = D \right] \leq \left( \frac{p^n}{D} \right) 2^D \left( 4k \right)^D \left( \frac{D}{n-p} \right)^D
\]

\[
\leq \left( \frac{e p h}{D} \right)^D 2^D \left( 4k \right)^D \frac{D}{n-p}^D
\]

\[
\leq \left( \frac{8 e p h}{1-p} \right)^D
\]

\[
= \left( c p h \right)^D
\]

\[
\text{Lemma: Let } P \text{ be a depth-} d \text{ formula where the number of gates at the top } d-1 \text{ levels is } s \text{ and the formula contains every gate at the bottom level } \leq \log s. \]

If \( P \) computes a parity of \( n \) variables,

\[
s \geq 2^{2^{\left( \frac{n}{d-1} \right)}}
\]

Case for \( d=2 \) was the exercise from last class.
Now for induction, restrict \( Y \) by \( p \). That leaves \( \frac{e^in}{\log s} \) variables unset, \( p = \frac{e^i}{\log s} \).

\( Y \) has \( \leq s \) CNF/DNF \( p_i \) that are \( \log s \)-CNF.

Probability that any of these have CDT \((p_i \land p) \leq \log s\) \( \leq e \cdot \frac{e^i}{\log s} \cdot \frac{1}{s} < \frac{1}{s} \)

so \( \exists p \) such that \( \forall p_i, \text{ CDT} (p \land p_i) \leq \log s \).

So replacing each CDT with DNF & blending at the next higher level, \( p \land p \) has \( \leq s \) gates \( \leq \log s \) at level \( d-1 \), and fan-in \( \leq \log s \) at level \( d-1 \).

Applying the inductive assumption with \( s, a-1, \frac{n'}{\log s} \):

\( n' \leq c \left( \frac{\log s}{\log s} \right)^{d-2} \)

so \( n \leq c \left( \frac{\log s}{\log s} \right)^{d-1} \)

Algorithmic application.

IN: \( k \)-CNF \( \phi \)

out: \# of \( \bar{x} \) s.t. \( \phi(\bar{x}) = \text{TRUE} \).

Exact randomized algorithm: partition, \( \@ \) random, variables into \( k \) groups: \( x_i = (1 - \frac{e^i}{k}) n \) \( \frac{1}{k} \).

\( |x_i| = \frac{e^i n}{k} \).
For each setting of the variable in \( a \), we look in the restriction \( p \).

Create \( \text{CDT}(p_{1p}) \) & count the # of solutions

Expected time: \( \text{exponential} \)

\[
\sum_0^\infty 0 \left( 1 \text{CDT} \left[ p_{1p} \right] \right) = 2^{n-pn} \text{Exp} \left( 0 \left( 1 \text{CDT} \right) \right)
\]