Last time, we discussed how a formula can be treated as a directed binary tree with variables or negated variables as the leaves. Given a circuit $c$, it has size $\text{size}(c)$ and depth $\text{depth}(c)$. Given a formula $F$, it has size $\text{size}(F)$ and depth $\text{depth}(F)$. Given a function $f$, $\text{csize}(f)$ is the size of the smallest circuit that computes $f$ and $\text{cdepth}(f)$ is the depth of the circuit with the lowest depth that computes $f$. Similarly define $\text{Fsize}(f)$ and $\text{Fdepth}(f)$ for formulas.

A key difference between circuits and formulas is that a gate in a circuit can have multiple outputs a gate in a formula can have only one output wire. The parity function illustrates the difference between formulas and circuits. Consider the parity function $f(x_1, x_2, \ldots, x_n) = x_1 \oplus x_2 \oplus \ldots x_n$. As a warm-up, $f(x_1, x_2) = (x_1 \land \bar{x}_2) \lor (\bar{x}_1 \land x_2)$ is a formula for $f$ with $n = 2$ attributes.

A general circuit $C_n$ for the parity function is given in Figure 1. $C_{n/2}$ and $C'_{n/2}$ represent a circuit for the parity function over the first and last $\frac{n}{2}$ literals (respectively). This circuit shows that the $\text{size}(C_n) = 2\text{size}(C_{n/2}) + 5$ so $\text{size}(C_n) = O(n)$.

However, a similar binary tree for a formula would require more copies of $F_{n/2}$, where $F_{n/2}$ is a formula tree for the parity function over the first and last $\frac{n}{2}$ literals. This is because the circuit given in Figure 1 uses the output of $C_{n/2}$ and $C'_{n/2}$ in two ways. So $\text{size}(F_n) = 4\text{size}(F_{n/2}) + 0$ as we need 4 copies of $F_{n/2}$ (two negated, two not) and this construction adds no additional literals. So $\text{size}(F_n) = O(n^2)$. We will actually show that this upper bound for the formula size is tight.

Note that, for all circuits $c$, $\text{size}(c) \leq 2^{\text{depth}(c)}$ however the converse is not true. Given a circuit $c$ is there an equivalent circuits $c'$ such that $\text{depth}(c')$ is approximately $\log \text{size}(c)$? We do not know answer to this question. However, we can show that it is indeed true for formulas. We can show that for every formula $F$, there is an equivalent formula $F'$ such that $\text{depth}(F') \simeq \log(\text{size}(F))$. We first start with the following result. Given a binary tree $T$ and node $x \in T$ and $T_x$ is the subtree rooted at $x$ and $|T_x|$ represents the size of $T_x$.

**Claim 1.** For any complete binary tree $T$, $\exists x \in T$ $\frac{1}{3}|T| \leq |T_x| \leq \frac{2}{3}|T|$.

![Figure 1. A general circuit construction for the parity function.](image-url)
Proof. Let \( x_1 \) represent the root of \( T \), \( x_2 \) be the root of larger subtree. That is, if \( x_1 \) has two children \( v_1 \) and \( v_2 \) then, if \(|T_{v_1}| \geq |T_{v_2}|\) then \( x_2 = v_1 \) else \( x_2 = v_2 \). Repeating this procedure, we can identify a sequence of nodes \( x_1, x_2, \ldots, x_k \) where \( x_k \) is a leaf.

Let \( i \) be the largest number such that \(|T_{x_i}| \leq \frac{|T|}{3}\). So \(|T_{x_{i-1}}| > \frac{|T|}{3}\) and therefore \(|T_{x_{i-1}}| \leq \frac{2|T|}{3}\) by our choice of \( x_i \) (that is, otherwise \( T_{x_i} \) was not the larger subtree of \( x_{i-1} \)). \( \square \)

**Theorem 1.** Let \( \phi \) be a formula. There is an equivalent formula \( \psi \) such that \( \text{depth}(\psi) \leq 1 + 2 \log \frac{3}{2}(\text{size}(\phi)) \).

Proof. Let \( m = \text{size}(\phi) \) and let \( T \) be the tree that represents the formula. By previous lemma there exist a node \( x \) such that \( \frac{m}{3} \leq |T_x| \leq \frac{2m}{3} \). \( T_x \) defines a subformula we call \( G \). \( F_0 \) is the formula resulting from placing 0 at \( x \) and \( F_1 \) the formula resulting from placing 1 at \( x \).

Let \( \psi = (F_0 \land \neg G) \lor (F_1 \land G) \). If \( G \) evaluates to 0, \( F_0 \) precisely captures \( \phi \). Likewise, \( F_1 \) captures \( \phi \) if \( G \) evaluates to 1. So \( \psi \) is equivalent to \( \phi \). Note that \( \text{depth}(\psi) = 2 + \max\{\text{depth}(F_0), \text{depth}(F_1), \text{depth}(G)\} \). Observe that by our choice of \( x \), sizes of formulas \( F_0 \), \( F_1 \) and \( G \) lies between \( m/3 \) and \( 2m/3 \).

We can solve for \( \text{depth}(\psi) \) using induction on formula size. The base case is clear. Assume that every formula \( F \) of size atmost \( m - 1 \) can be converted into a formula of depth \( 1 + 2 \log \frac{3}{2}(\text{size}(F)) \). Now note that

\[
\text{depth}(\psi) = 2 + \max\{\text{depth}(F_0), \text{depth}(F_1), \text{depth}(G)\} \\
\leq 2 + 1 + 2 \log \frac{3}{2} \left( \frac{2m}{3} \right) \\
= 2 + 1 + 2 \left[ \log \frac{3}{2} + \log \frac{m}{3} \right] \\
= 1 + 2 \log \frac{3}{2}(m).
\]

\( \square \)

**Corollary 1.** For every boolean function \( f \), \( \text{FSize}(f) = \theta(2^{F\text{Depth}(f)})\).

We now introduce a method called the Random Restriction Method that helps us to prove some lower bounds on formula size. Given a formula in \( n \)-variables, randomly assign value to some of the variables. How does the resulting formula look like? We will show that the resulting formula size must go down considerably. From here on, we will use \( n \) as the number of variables.

**Definition 1.** A partial assignment to \( \{x_1, \ldots, x_n\} \) is a string over \( \{0, 1, *\}^n \). Given a function \( f \) and partial assignment \( a \), \( f \) restricted to \( a \) (\( f|_a \)) denotes the function obtained by applying \( a \) to \( f \).

The semantics of a partial assignment is that a 0 or 1 at the \( i^{th} \) position is interpreted as an assignment of \( x_i = 0 \) or 1 while a * indicates that \( x_i \) remains unassigned. If \( m \) variables are unassigned in the partial assignment \( a \), then \( f|_a \) is a function from \( \{0, 1\}^m \) to \( \{0, 1\} \).

**Definition 2.** \( R_k \) is the set of all partial assignments leaving \( k \) variables unassigned. That is, \( R_k = \{a \in \{0, 1, *\}^n \mid \text{number of *s in } a \text{ is } k \} \).
Consider the process of uniformly at random picking a partial assignment from $R_k$. This can be done by first uniformly at random picking $(n - k)$ variables and assigning a random value from \{0, 1\} to each of them. This is process is same the following process: Uniformly at random pick an unassigned variable and uniformly at random assign it a value of 0 or 1. Repeat this process $n - k$ times. The following theorem is due to Subbotovskaya.

**Theorem 2.** For every boolean function $f$ and $k > 0$

$$\Pr_{a \in R_k} \left[ \text{FSize}(f|_a) \leq 4 \left( \frac{k}{n} \right)^{3/2} \text{FSize}(f) \right] \geq \frac{3}{4}.$$  

*Proof.* Let $\phi$ be a minimal formula for $f$ with $\text{size}(\phi) = s$. As mentioned before, picking a random assignment can be thought as a stage by stage process of picking a (unassigned) variable and giving it a value of zero or one. We analyze the effect of the random restriction in stage by stage process.

Consider the first stage. This stage randomly picks a variable $v \in \{x_1, \ldots, x_n\}$ and randomly set its value to 0 or 1. Let $\phi_v$ denote the resulting formula. Let the random variable $X$ be the amount by which the size of $\phi$ goes down by this process. More precisely the random variable $X$ is $\text{size}(\phi) - \text{size}(\phi_v)$. Let $Y$ be a random variable that denotes the number of times $v$ appears in $\phi$. Observe that $X \geq Y$, and thus $E(X) \geq E(Y)$. What is the expectation of $Y$?

Let $a_i$ represent the number of times the variable $x_i$ appears in $\phi$. So

$$E[Y] = \Pr[v = x_1]a_1 + \Pr[v = x_2]a_2 + \ldots + \Pr[v = x_n]a_n = \frac{\sum a_i}{n} = \frac{s}{n}.$$  

Thus $E(X) \geq s/n$. We will now show that this expectation is indeed a bit higher.

We use $\ell_i$ to denote either $x_i$ or $\overline{x_i}$. Given a variable $x_i$, let $S(x_i)$ be a structure of the following form.

```
  g
    \_ \\
   \_2
  \_1
```

We first show that neither $x_i$ nor $\overline{x_i}$ can occur in $G$. Let us first assume that $\ell_i = x_i$. Let us also assume that the node is an $\land$ node. Say $x_i$ appears in $G$. Now consider the formula $\phi'$ obtained by replacing all occurrences of $x_i$ in $G$ by 1. Certainly the size of $\phi$ is less than the size of $\phi$. We now claim that $\phi'$ is equivalent to $\phi$. Consider any assignment in which $x_i$ is 1. We replaced $x_i$ with 1 in $G$, so this has no effect. So $\phi$ and $\phi'$ evaluate to the same value on this assignment. Now consider an assignment in which $x_i$ is zero. If $x_i = 0$, then $x_i \land G' = 0$ for any formula $G'$. So both $\phi$ and $\phi'$ evaluate the same value. However assumed
φ is the smallest formula that computes our function f. Other cases, where \( x_i \) appears in G, \( \ell_i = \overline{x_i} \), and the gate g is an or-gate can be handled similarly.

Clearly, the number of times \( S(x_i) \) equals the number of times \( \ell_i \) appears in the formula. If the gate g in \( S(x_i) \) is an AND-gate, and if a random assignment gives a value zero to the literal \( \ell_i \), then we can replace the entire structure \( S(x_i) \) by zero. Similarly if the gate g is an OR-gate, then the random assignment gives a value 1 to \( \ell_i \), then also we can replace the entire structure by 1.

Recall that our random restriction uniformly picked a variable \( v \) and chose a random value for it. Let \( Z \) denote the random variable, that denotes the number of structure \( S(v) \) that can be replaced by zero or 1. By previous discussion, the expectation of \( Z \) equals half of (expected) number of times \( v \) appears. Thus \( E(Z) = E(Y) / 2 \). Thus the formula size of \( \phi_v \) further goes down by \( E(Z) / 2 \) in expectation. Thus

\[
E(X) \geq \frac{s}{n} + \frac{s}{2n} = \frac{3s}{2n}.
\]

Thus

\[
E[\text{size}(\phi|_v)] \leq s - \frac{3s}{2m} = s\left(1 - \frac{3}{2} \cdot \frac{1}{n}\right) \leq s\left(1 - \frac{1}{n}\right)^{3/2}.
\]

The formula \( \phi_v \) is a formula in \( n - 1 \) variables. After randomly choosing second variable, the size goes down by a factor of \( (1 - \frac{1}{n-1})^{3/2} \). Thus

\[
E[\text{Size}(\phi|_a)] \leq s \left(1 - \frac{1}{n}\right)^{3/2} \left(1 - \frac{1}{n-1}\right)^{3/2} \ldots s \left(1 - \frac{1}{k+1}\right)^{3/2}
\]

\[
\leq \left(\frac{k}{n}\right)^{3/2} s.
\]

By Markov’s Inequality, for any random variable X, \( \Pr(X > vE(X)) \leq \frac{1}{v} \) so \( \Pr(X \leq vE(X)) \geq 1 - \frac{1}{v} \). The theorem follows by setting \( v = 4 \).

We can use above theorem to obtain a lowerbound on the formula complexity of the parity function. Let \( \oplus_n \) denote the parity of \( n \) variables. Clearly \( \oplus_1 \) in the above theorem, we obtain

\[
\text{FSize}(\oplus_n) \leq 4 \cdot \frac{1}{n^{3/2}} \text{FSize}(\oplus_1).
\]

Thus \( \text{FSize}(\oplus_n) \geq \frac{1}{4} n^{3/2} \).