This lecture investigates the circuit-SAT problem. We will see that improved algorithms for circuit-SAT can yield lower-bounds.

**Definition 1.** Circuit-SAT: Given a circuit $C$, is there any input $x$ such that $C(x) = 1$?

If Circuit-SAT is in $P$, then $P = NP$, since Circuit-SAT is NP-hard. This implies that the polynomial hierarchy collapses to P. To see why this is true, consider this statement from the 3rd level of the polynomial hierarchy, where $R$ is polynomially decidable and $x$ is fixed:

$$x \in L \iff \exists y_1 \forall y_2 \exists y_3 R(x, y_1, y_2, y_3)$$

So, given $y_1$ and $y_2$, we can see that

$$\exists y_3 R(x, y_1, y_2, y_3) \in NP = P$$

We then find a polynomially computable relation $S$, such that

$$\exists y_3 R(x, y_1, y_2, y_3) = S(x, y_1, y_2)$$

Therefore

$$x \in L \iff \exists y_1 \forall y_2 S(x, y_1, y_2, y_3)$$
So, given $y_1$, we can see that

$$\forall y_2 S(x, y_1, y_2) \in \text{Co-NP} = \text{NP} = P$$

We then find a polynomially-computable relation $T$ such that

$$S(x, y_1, y_2) = \exists y_2 T(x, y_2)$$

Thus, the language $L$ is in $P$. Looking this example, we can see that if $P = \text{NP}$ then the polynomial hierarchy collapses. So, by contraposition, we need only prove that two layers of the hierarchy are distinct in order to show that $P \neq \text{NP}$.

We will now review Meyer’s Theorem, which states that if $\text{EXP} \subseteq P/\text{Poly}$, then $\text{EXP} \subseteq \Sigma_2^P$.

**Definition 2.** A Circuit is **locally computable** if, given $n$ inputs bits, the name $i$ of a gate, and the names $k(i)$ and $j(i)$ of input gates, we can compute $op_i(k(i), j(i))$ (the output of gate $i$) in time $\text{poly}(\text{len}(i) + n)$.

**Theorem 1** (Hennie, Stearns). For any time $T(n)$ algorithm on a turing machine, there is a size $O(T(n)\log(T(n)))$ locally computable circuit simulating the algorithm.

We are now ready to proof Meyer’s theorem.

**Theorem 2** (Meyers). If $\text{EXP} \subseteq P/\text{Poly}$, then $\text{EXP} \subseteq \Sigma_2^P$.

**Proof.** Assume $\text{EXP} \subseteq P/\text{Poly}$. Take $L \in \text{EXP}$. Let $C_n$ be a locally computable circuit that decides $L$ on $n$-bit inputs. Let $L'$ be the language that maps a given gate $i$ of $C_n$ and input $x$ to the value of gate $i$ on $x$. Since $L \in \text{EXP}$, we know $L' \in \text{EXP}$, so $L' \in P/\text{Poly}$. So there exists a circuit $C''(x, i)$ deciding $L'$. Therefore

$$x \in L \iff \exists C'' \forall i [op_i(c''(x, j(i)), c''(x, k(i))) = c''(x, i) \land c''(x, \text{output\_gate}) = 1]$$
Since this formula is of the form $\exists \forall \psi$, with polynomially-computable $\psi$, we know that $L \in \Sigma^P_2$. 

Using Meyer’s theorem, we can investigate what happens if we just have a “pretty good” algorithm for circuit-SAT. We will explore this idea in the following theorem.

**Theorem 3.** If $\text{circuit-SAT} \in \text{Time}(2^{n^{o(1)}})$, then $\text{NEXP} \not\subseteq \text{P/Poly}$

**Proof.** Assume $\text{circuit-SAT} \in \text{Time}(2^{n^{o(1)}})$. We know either $\text{EXP} \subseteq \text{P/Poly}$ or $\text{EXP} \not\subseteq \text{P/Poly}$. If $\text{EXP} \not\subseteq \text{P/Poly}$, since $\text{EXP} \subseteq \text{NEXP}$, the statement holds.

If $\text{EXP} \subseteq \text{P/Poly}$, then $\text{EXP} = \Sigma^P_2$ (this is Meyer’s theorem).

So, if we choose and arbitrary $L \in \text{EXP}$, we get that there is some polynomially-computable relation $S$ such that:

$$x \in L \leftrightarrow \exists y_1 \forall y_2 S(x, y_1, y_2)$$

Since circuit-SAT is in sub-exponential time, we can use the same algorithm for circuit-SAT to compute the complement of circuit-SAT. Therefore, the formula $\forall y_2 S(x, y_1, y_2)$ is in $\text{Time}(2^{n^{o(1)}})$. Therefore, $L \in \text{NTIME}(2^{n^{o(1)}})$. So $\Sigma^P_2 \subseteq \text{EXP} \subseteq \text{NTIME}(2^{n^{o(1)}})$. By a padding argument, we can see that $\exists T \in \text{Time}(n^{\omega(1)})$ such that $\Sigma^{T(n)}_3 \subseteq \text{NEXP}$. So since $\Sigma^{T(n)}_3 \not\subseteq \text{P/Poly}$, $\text{NEXP} \not\subseteq \text{P/Poly}$. 

Given a circuit $C$, the naive approach to circuit-SAT can try all possible inputs on the circuit in time $|C|^2^n$. The following theorem, due to Ryan Williams, shows that slight improvements to this naive approach yields the same circuit lower bound.

**Theorem 4.** If $\text{Circuit-SAT} \in \text{Time}(|C|^2^n/n^{\omega(1)})$, then $\text{NEXP} \not\subseteq \text{P/Poly}$.

**Proof.** Assume that $\text{Circuit-SAT} \in \text{Time}(|C|^2^n/n^{\omega(1)})$. We define can define the problem circuit-TAUT to determine whether a Circuit returns 1 on all inputs (i.e., is a tautology). Let $\neg C$ denote the circuit $C$ with a negation gate applied to it’s final output.
Since \( \text{circuit-TAUT}(C) = \neg\text{circuit-SAT}(\neg C) \), our initial hypothesis implies that \( \text{circuit-TAUT} \in \text{TIME}(|C|2^n/n^{\omega(1)}) \). We will show that this implies that \( \text{NEXP} \not\subseteq \text{P/Poly} \).

Assume that \( \text{NEXP} \subseteq \text{P/Poly} \) and \( \text{circuit-TAUT} \in \text{TIME}(|C|2^n/n^{\omega(1)}) \). We will use these assumptions to contradict the nondeterministic hierarchy theorem. Let \( L \) be a language that is exactly in \( \text{NTIME}(2^n) \). In other words, there is a relation \( R \) computable in time \( 2^n \) such that:

\[
x \in L \iff \exists y, |y| = 2^n \land R(x, y)
\]

By the theorem of Hennie and Stearns, there is a locally computable circuit, \( C_R \), which computes \( R \). We know \( |C_R| = O(2^n n) \). We can then write the formula:

\[
x \in L \iff \exists g_1 \ldots g_{2^n n} \text{s.t.} \text{“the value of each gate follows from it’s inputs”}
\]

So the values \( g_1 \ldots g_{2^n n} \) act as a transcript of \( C_R \). Last class, we saw the easy witness lemma. This states that \( \text{NEXP} \subseteq \text{P/Poly} \) if and only if every positive instance of an \( \text{NEXP} \) problem has a succinctly describable witness (i.e., describable as a poly-size circuit computing the \( i^{th} \) bit of the witness). We can see that the previous formula is in \( \text{NEXP} \) so there must be a succinct witness \( C'' \). In other words:

\[
x \in L \iff \exists C'' \forall i = 1 \ldots 2^n n, op_i(C''(j(i)), C''(k(i))) = C''(i) \land C''(x, output\_gate) = True
\]

We define \( T_{C''} \) to be the circuit on \( n + \log(n) \) inputs that computes \( T_{C''}(i) = op_i(C''(k(i)), C''(j(i))) = C''(i) \). This gives the formula:

\[
x \in L \iff \exists C'', T_{C''} \text{is a tautology}
\]

By out initial assumption, this implies that \( L \in \text{NTIME}(2^{n+\log(n)+O(1)}/n^{\omega(1)}) = \text{NTIME}(2^n/n^{\omega(1)}) \).
So \( \text{NTIME}(2^n) = \text{NTIME}(o(2^n)) \). But this contradicts the non-deterministic time hierarchy theorem, completing the proof.

We will conclude by stating another theorem by Ryan Williams, along with some corollaries.

**Theorem 5.** For all depths \( \delta \) there is an \( \epsilon \) such that \( \text{ACC}_6\text{-SAT} \in \text{TIME}(2^{n-n^\epsilon}) \).

**Lemma 1.** If \( C \) is a class of circuits such that \( C\text{-TAUT} \in \text{NTIME}(2^n/n^{\omega(1)}) \), and \( P \in C \), then \( \text{circuit-TAUT} \in \text{NTIME}(2^n/n^{\omega(1)}) \).

**Corollary 1.** If \( C\text{-TAUT} \in \text{NTIME}(2^n/n^{\omega(1)}) \), then \( \text{NEXP} \not\subseteq C \).

**Corollary 2.** \( \text{NEXP} \not\subseteq \text{ACC}_6 \)