In this lecture, we address the question: Does SAT has small (i.e., polynomial-sized) circuits? In other terms, is $NP \subseteq P/poly$?

We do not have a definite answer to this question. In relation to this question, we have instead the following theorem:

**Theorem**: $NP \subseteq P/poly \implies PH = \Sigma_2^P$

**Proof**:

**Facts**

(i) SAT is self-reducible. *Explanation*: There are many definitions of self-reducibility. One of them, is the one that follows:

*Definition*: $A$ is self-reducible if $A \in P^A$, via an oracle TM that on input $x$, ask queries only about strings $y < x$ (lexicographic order).

SAT satisfies this notion of self-reducibility. We give the following algorithm as a proof of this statement:

on input $\phi$

if $\phi$ has NO variables

then return 1 if $\phi$ evaluates to 1

and return 0 if $\phi$ evaluates to 0

else

let $x$ be a variable in $\phi$

let $\phi_1$ be the result of replacing $x$ by 1

let $\phi_0$ be the result of replacing $x$ by 0

if $\phi_1 \in SAT$, then return 1

if $\phi_0 \in SAT$, then return 1

else return 0

We will use this fact (i) to give a $\Sigma_2^P$ algorithm for $\Pi_2^P$ ($= co-NP^{NP}$). That is, let $A \in \Pi_2^P$ iff $A$ is recognized by a co-NP machine $M$ with oracle SAT. We will show: $A \in \Sigma_2^P$. Below, is the proof of this last statement:

**Proof**:

on input $x$ (with $|x| = n$)

(note that $M$ asks queries only of inputs of length $\leq |x|^k$ for some $k$.)

existentially guess a sequence of circuits $C_1, C_2, \ldots, C_n$ using universal moves, do the following:

1) Simulate $M^{Sat}(x)$ (answering oracle queries using $C_1, C_2, \ldots, C_n$)

2) Verify that each of the circuits $C_1, C_2, \ldots, C_n$ is computing SAT correctly.

That is for all strings $\phi$ with $|\phi| \leq n^k$, $C_{\phi}(\phi) = 1$ , iff :
\[ \phi \text{ has NO variables and evaluates to TRUE} \quad OR \\
\left( C_{\phi_0}(\phi_0) = 1 \ OR \ C_{\phi_1}(\phi_1) = 1 \right) \]. \ \Box

What remains to prove our theorem is this statement:

\[ \Pi_2^p = \Sigma_2^p \implies PH = \Sigma_2^p \]

Proof: \textit{(by induction)}

Let \( A \in \Sigma_2^p \) then \( A \) is recognized by an ATM that begins in existential states then goes to universal states and the goes back to existential states.

Note that:
\( B = \{ (C, x) : C \text{ is a universal configuration of } M \text{ that is accepting on input } x \} \); then:
\( B \in \Pi_2^p \subseteq \Sigma_2^p \).

Here is the \( \Sigma_2^p \) algorithm for \( A \):

\begin{itemize}
  \item on input \( x \)
  \item existentially simulate \( M(x) \) until it reaches a universal configuration \( C \), then use the \( \Sigma_2^p \) algorithm to see if \( (C, x) \in B \).
\end{itemize}

\( \text{(Inductive part of the proof):} \)

We have \( \Sigma_3^p \subseteq \Sigma_2^p \). Assume \( \Sigma_i^p \subseteq \Sigma_2^p \). We will show: \( \Sigma_{i+1}^p \subseteq \Sigma_2^p \).

\( \Sigma_{i+1}^p = NP_{\Sigma_i^p} \subseteq NP_{\Sigma_2^p} = \Sigma_3^p = \Sigma_2^p \). \ \Box