Boolean Circuits and the Karp-Lipton Theorem

This material was presented in class on February 23, 2016.

Before presenting the proof of the Karp-Lipton Theorem we covered Theorem 2.18 and Definitions 6.1, 6.2, and 6.5. These items are all presented clearly in the textbook and won’t be repeated here.

Karp-Lipton Theorem: If \( \text{NP} \subseteq \text{P}/\text{poly} \), then \( \text{PH} = \Sigma_2^P \).

Proof: It suffices to show that, if \( \text{NP} \subseteq \text{P}/\text{poly} \), then \( \Pi_2 \text{SAT} \in \Sigma_2^P \).

Recall that \( \Pi_2 \text{SAT} \) consists of all true QBFs of the form

\[
∀u \in \{0, 1\}^n \exists v \in \{0, 1\}^n \phi(u, v) = 1,
\]

where \( \phi \) is a quantifier-free boolean formula on \( 2^n \) variables with \( m \) clauses.

Note that (1) is of the form \( ∀u \in \{0, 1\}^n \text{SAT} \); that is, for any fixed \( \phi \) and \( u \), the part of (1) that begins with \( ∃ \) is just \( ∃v \in \{0, 1\}^n φ_u(v) = 1 \), where \( φ_u(·) \) is the formula \( φ(·, ·) \) with the first \( n \) boolean variables instantiated as in \( u \) and the last \( n \) boolean variables left free. This is, of course, a SAT instance.

Our hypothesis is that \( \text{SAT} \in \text{P}/\text{poly} \). So there is a polynomial \( p \) and a \( p(n, m) \)-sized circuit family \( \{C_{n,m}\} \) such that

\[
∀\phi, u \; C_{n,m}(\phi, u) = 1 \iff ∃v \in \{0, 1\}^n φ_u(v) = 1.
\]

Here, “\( C_{n,m}(\phi, u) \)” means “the circuit \( C_{n,m} \) evaluated on the SAT instance determined by \( \phi \) and \( u \).”

Recall that there is a polynomial-sized circuit family \( \{C'_{n,m}\} \) that reduces the search problem for SAT to the decision problem for SAT. Given an oracle that decides SAT, a circuit \( C'_{n,m} \) can produce an assignment that satisfies a formula, provided such an assignment exists. Whenever \( C'_{n,m} \) needs to make an oracle call on a \( k \)-variable, \( ℓ \)-clause formula and feed the answer to a gate \( g \), it can instead feed that formula to \( C_{k,ℓ} \) and feed the output to \( g \). There will be a polynomial number \( q(n) \) of such calls, the sizes \( (k_1, ℓ_1), \ldots, (k_{q(n)}, ℓ_{q(n)}) \) are all polynomial in \( (n, m) \), and the circuits \( C_{k_i,ℓ_i} \) are of size polynomial in \( k_i \) and \( ℓ_i \). Therefore, under the hypothesis that \( \text{SAT} \in \text{P}/\text{poly} \), we can “compose” these circuit families \( \{C_{n,m}\} \) and \( \{C'_{n,m}\} \) to get a polynomial-sized circuit family \( \{D_{n,m}\} \) that, given a SAT instance as input, produces a satisfying assignment if one exists. (We need the hypothesis to assert the existence of \( \{C_{n,m}\} \) but not to assert the existence of \( \{C'_{n,m}\} \).) Let \( w(n, m) \) be the (polynomial) number of bits needed to encode \( D_{n,m} \). Denote by \( D_{n,m}(\phi, u) \) the output of \( D_{n,m} \) on the formula \( φ_u \) determined by \( \phi \) and \( u \).

Now consider the following \( \Sigma_2^P \) expression:

\[
∃D_{n,m} \in \{0, 1\}^{w(n, m)} ∀u \in \{0, 1\}^n \; φ_u(D_{n,m}(\phi, u)) = 1.
\]

We have just argued that, if (1) is true and \( \text{NP} \subseteq \text{P}/\text{poly} \), then (2) is true. On the other hand, if (1) is false, then (2) is also false, regardless of whether \( \text{NP} \subseteq \text{P}/\text{poly} \). Thus, under the assumption that \( \text{NP} \subseteq \text{P}/\text{poly} \), the \( \Pi_2 \text{SAT} \) formula (1) is equivalent to the \( \Sigma_2^P \) expression (2).