CS 517: PH in terms of Oracle Classes

Writeup by Mike Rosulek

Theorem 1 *For $k \geq 1$, $\Sigma_k = NP^{\Sigma_{k-1}}$*

Proof ($\subseteq$) Suppose we are given an arbitrary language $L \in \Sigma_k$. That is, L can be written as

$$L = \{x \mid \exists w_1 \forall w_2 \cdots \mathrm{Q} w_k : M(x, w_1, \ldots, w_k) = 1\}$$

for some polytime M. We want to show that L can be written as an $\mathsf{NP}^{\Sigma_{k-1}}$ language.

Define the related language:

$$L' = \{(x, w_1) \mid \exists w_2 \forall w_3 \cdots \mathrm{Q} w_k : M(x, w_1, \ldots, w_k) = 0\}$$

This language is clearly in Σ_{k-1}. Now observe:

$$\begin{aligned} x \in L &\iff \exists w_1 \forall w_2 \cdots \mathrm{Q} w_k : M(x, w_1, \ldots, w_k) = 1 \\ &\iff \exists w_1 \neg[\exists w_2 \forall w_3 \cdots \mathrm{Q} w_k : M(x, w_1, \ldots, w_k) = 0] \\ &\iff \exists w_1 : (x, w_1) \notin L' \end{aligned}$$

Hence we can rewrite L in an equivalent way, $L = \{x \mid \exists w_1 : (x, w_1) \notin L'\}$. We have written L in terms of an existential quantifier followed by a condition that can be verified in polynomial time with an L'-oracle. Hence, $L \in \mathsf{NP}^{L'} \subseteq \mathsf{NP}^{\Sigma_{k-1}}$.

($\supseteq$) Suppose we are given an arbitrary language $L \in \mathsf{NP}^{\Sigma_{k-1}}$. That is, L can be written as

$$\begin{aligned} L &= \{x \mid \exists w^* : M^A(x, w^*) = 1\}, \text{ where} \\ A &= \{q \mid \exists w_1 \forall w_2 \cdots \mathrm{Q} w_{k-1} : T(q, w_1, \ldots, w_{k-1}) = 1\} \end{aligned}$$

for some poly-time M and T. We want to show that L can be written as a Σ_k language.

We first introduce some helpful terminology. Let C be some oracle TM and let O be some oracle. We can think of the computation of $C^O(x)$ as an *interaction*, where C repeatedly sends a *query* q to O, and O sends a *response* $r \in \{0, 1\}$ back to C. Let $\boldsymbol{t} = (q_1, r_1, \ldots, q_n, r_n)$ denote the **transcript** of such an interaction. We say that $\boldsymbol{t}$ is:

- **consistent with caller** $C(x)$ **accepting** if, when running on input x, the oracle TM C will indeed make the sequence of queries $q_1, \ldots, q_n$ and finally accept, as long as it receives responses $r_1, \ldots, r_n$ from those queries.
- **consistent with oracle** O if for all i: $r_i = 1 \Rightarrow q_i \in O$ and $r_i = 0 \Rightarrow q_i \notin O$. ■

Using this terminology, we can restate what it means for an oracle machine to accept a string:

$$\begin{aligned} C^O(x) = 1 \iff \exists \boldsymbol{t} : \boldsymbol{t} &\text{ is consistent with caller } C(x) \text{ accepting} \\ \text{and } \boldsymbol{t} &\text{ is consistent with oracle } O \end{aligned}$$

We can now rewrite the language L in terms of this definition, and expand:

$$
\begin{aligned}
x \in L \iff\ & \exists w^* : M^A(x, w^*) = 1 \\
\iff\ & \exists w^*, \boldsymbol{t} : \boldsymbol{t} \text{ is consistent with caller } M(x, w^*) \text{ accepting} \\
& \qquad\qquad \text{and } \boldsymbol{t} \text{ is consistent with oracle } A \\
\iff\ & \exists w^*, \boldsymbol{t} = (q_1, r_1, \ldots, q_n, r_n) : \\
& \qquad\quad \boldsymbol{t} \text{ is consistent with caller } M(x, w^*) \text{ accepting} \\
& \qquad\quad \text{and } [r_1 = 1 \Rightarrow q_1 \in A] \text{ and } [r_1 = 0 \Rightarrow q_1 \notin A] \\
& \qquad\qquad \vdots \\
& \qquad\quad \text{and } [r_n = 1 \Rightarrow q_n \in A] \text{ and } [r_n = 0 \Rightarrow q_n \notin A] \\
\iff\ & \exists w^*, \boldsymbol{t} = (q_1, r_1, \ldots, q_n, r_n) : \\
& \qquad\quad \boldsymbol{t} \text{ is consistent with caller } M(x, w^*) \text{ accepting} \\
& \qquad\quad \text{and } r_1 = 1 \Rightarrow [\exists w_{1,1} \forall w_{1,2} \cdots \mathrm{Q} w_{1,k-1} : T(q_1, w_{1,1}, \ldots, w_{1,k-1}) = 1] \\
& \qquad\quad \text{and } r_1 = 0 \Rightarrow [\forall w'_{1,1} \exists w'_{1,2} \cdots \mathrm{Q} w'_{1,k-1} : T(q_1, w'_{1,1}, \ldots, w'_{1,k-1}) = 0] \\
& \qquad\qquad \vdots \\
& \qquad\quad \text{and } r_n = 1 \Rightarrow [\exists w_{n,1} \forall w_{n,2} \cdots \mathrm{Q} w_{n,k-1} : T(q_n, w_{n,1}, \ldots, w_{n,k-1}) = 1] \\
& \qquad\quad \text{and } r_n = 0 \Rightarrow [\forall w'_{n,1} \exists w'_{n,2} \cdots \mathrm{Q} w'_{n,k-1} : T(q_n, w'_{n,1}, \ldots, w'_{n,k-1}) = 0] \\
\iff\ & \exists w^*, \boldsymbol{t} = (q_1, r_1, \ldots, q_n, r_n), w_{1,1}, \ldots, w_{n,1} : \\
& \forall w_{1,2}, \ldots, w_{n,2}, w'_{1,1}, \ldots, w'_{n,1} : \\
& \exists w_{1,3}, \ldots, w_{n,3}, w'_{1,2}, \ldots, w'_{n,2} : \\
& \quad \vdots \\
& \mathrm{Q} w'_{1,k-1}, \ldots, w'_{n,k-1} : \\
& \qquad\quad \boldsymbol{t} \text{ is consistent with caller } M(x, w^*) \text{ accepting} \\
& \qquad\quad \text{and } [r_1 = 1 \Rightarrow T(q_1, w_{1,1}, \ldots, w_{1,k-1}) = 1] \\
& \qquad\quad \text{and } [r_1 = 0 \Rightarrow T(q_1, w'_{1,1}, \ldots, w'_{1,k-1}) = 0] \\
& \qquad\qquad \vdots \\
& \qquad\quad \text{and } [r_n = 1 \Rightarrow T(q_n, w_{n,1}, \ldots, w_{n,k-1}) = 1] \\
& \qquad\quad \text{and } [r_n = 0 \Rightarrow T(q_n, w'_{n,1}, \ldots, w'_{n,k-1}) = 0]
\end{aligned}
$$

This final expression consists of k alternating quantifiers (beginning with $\exists$), followed by a condition that can be checked in polynomial time. This shows that $L \in \Sigma_k$, as desired.