12 Turing Machines with Oracles

Let $A \subseteq \{0,1\}^\star$ be any decision problem.

A TM, M with oracle A, written M^A has an **oracle tape**. When it writes w on its oracle tape and enters a query state, it finds out on the next step whether $w \in A$.

Let P^A, NP^A, etc., be the complexity classes of polynomial-time TM's with oracle, A, nondeterministic polynomial-time TM's with oracle, A, etc.

The **idea** of using oracles, is that most of the proof methods we have been using tend to go through when the same oracle, A, is given to all TM's being considered. We say that a statement about Turing Machines, S, **relatives**, if for all $A \subseteq \{0,1\}^\star$, $S \Leftrightarrow S^A$, where S^A is the statement S where all TM's are given the oracle A. Most theorems we have proved so far in 601 relativize.

The main result of today is:

Theorem 12.1 *[Baker, Gill and Solovay] There exist oracles A, B,* s.t. $\text{P}^A = \text{NP}^A$ *and* $\text{P}^B \neq \text{NP}^B$.

Corollary 12.2 *The statements "P = NP" and "*$\text{P} \neq \text{NP}$*" do not relativize.*

Proof: Let $A = \text{QSAT}$, or any other PSPACE-complete problem.

Claim 12.3 $\text{P}^{\text{QSAT}} \subseteq \text{NP}^{\text{QSAT}} = \text{PSPACE}$

To prove Claim 12.3, just observe that $\text{PSPACE} \subseteq \text{P}^{\text{QSAT}}$ and $\text{NP}^{\text{QSAT}} \subseteq \text{PSPACE}$.

We construct the oracle, B, by hiding useful information among the 2^n strings of length n such that an NP machine can guess and find it, but a P machine cannot.

We now construct an oracle B s.t. , $\mathrm{P}^B \neq \mathrm{NP}^B$.

Let the undecidable problem UHALT $\stackrel{\text{def}}{=} \{1^n \mid n \in \text{HALT}\}$.

We will construct $B \subseteq \{0,1\}^\star$ to have at most one string of each length, so that

$$\text{UHALT} \quad = \quad \{1^n \mid \exists w \in B\ (|w| = n)\}\ .$$

That is, B will have a string of length n iff $n \in$ UHALT. It thus follows that UHALT $\in \mathrm{NP}^B$.

However, we will hide these strings in such a way that for each unary language accepted by PTIME TM, M_i, $M_i(1^r)^B$ receives a "yes" on some query for at most finitely many r. That is, all but finitely many of the strings in B are hidden from each fixed unary languages in P^B. Thus a unary language is in P iff it is in P^B, so UHALT $\notin \mathrm{P}^B$.

To construct B, let $C_0, C_1, C_2, \ldots$ be the set of **clocked, ptime TM's** where C_i simultates M_i, but also keeps a clock that runs for at most in^i steps for all inputs, w, of length n. If the clock runs out before $M_i(w)$ halts, then $C_i(w)$ halts and rejects.

We construct B inductively. Assume that B_n is the initial segment of B consisting of strings of length $< n$ and that E_n is a set of fewer than $2^{(n-1)}$ strings that have been previously excluded, i.e., we will never put them into B.

For a **total** of 2^{n-1} steps, until you run out of time, for each $i = 0, 1, \ldots \infty$, for each unary input, 1^j, $j = 1, 2, \ldots n$, simulate $C_i^{B_n}(1^j)$. Let E_{n+1} be E_n together with all the queried strings w of length at least n. These strings were not in B_n when queried by $C_i^{B_n}(1^j)$, and they will be kept forever out of B, so all these simulations remain unchanged when we add strings to B.

By construction, E_{n+1} contains fewer than 2^n strings. Let $w_n \in \{0,1\}^n - E_{n+1}$.

If $n \in$ HALT, let $B_{n+1} = B_n \cup \{w_n\}$; otherwise, let $B_{n+1} = B_n$.

Why does this construction work? Note, that for each fixed i, it takes time less than $i^2 n^{i+1}$ to complete the simulation of $C_k(1^j)$ for all $k \leq i$ and $j \leq n$. Let N_i be such that $2^{N_i - 1} > i^2 N_i^{i+1}$. Thus, C_i^B never gets a positive answer to a query concerning a string of length greater than N_i. □