The Time-Hierarchy Theorems

This material was presented in class on Feb 2, 2016. In the latter part of the Feb 2, 2016, lecture, we covered some definitions and notation needed to state and prove the Baker-Gill-Solovay Theorem; they can be found in the lecture notes for Feb 4, 2016.

Some technical points that were covered in class on Feb 2 are not explained fully in these notes but instead marked by **"(WHY?)."** You are encouraged to think these points through in detail (and to ask questions if they're not clear to you).

Recall that, in the Jan 28, 2016, lecture, we used diagonalization to prove that the sets UC and HALT are undecidable. In fact, diagonalization can be used to prove some basic facts about computational complexity, not just decidability. Unfortunately, there is no concise definition of the term "diagonalization" that is suitable for this course. However, Section 3.4 of your textbook gives the following characterization that will suffice for our purposes: We will use the term to describe any technique that relies solely upon the following properties of TMs:

I The existence of an effective representation of TMs by strings

II The ability of one TM to simulate any other without much overhead in running time

An example of an "effective representation" is the one given in Figure 1.7 (a corrected version of which is included in the notes from Jan 28). An example of a simulation that does not require much overhead in running time is the one given in the proof of Theorem 1.9. Diagonalization can be used to prove the

Time-Hierarchy Theorem: If f and g are time-constructible functions satisfying $f(n) \log f(n) = o(g(n))$, then $\mathrm{DTIME}(f) \subsetneq \mathrm{DTIME}(g)$.

The following weaker version of this theorem is sufficient for purposes of this class. In the proof, $\mathcal{U}$ refers to the universal TM of Theorem 1.9.

Theorem 1 *For any positive integer d,* $\mathrm{DTIME}(n^d) \subsetneq \mathrm{DTIME}(n^{d+1})$.

Proof. Consider D, a TM that proceeds as follows: On input x, D simulates $\mathcal{U}(x, x)$ for $|x|^{d+0.4}$ steps. If $\mathcal{U}$ halts within this number of steps and outputs a bit b, then output $1 - b$. Else, output 0.

Note that, on any input x, D halts within $|x|^{d+0.4} \leq |x|^{d+1}$ steps and outputs 0 or 1. Thus $L(D)$ is well defined and in $\mathrm{DTIME}(n^{d+1})$. We must show that it is not in $\mathrm{DTIME}(n^d)$.

Assume, by way of contradiction, that there is a TM M and a constant c such that, for all $x \in \{0, 1\}$, M halts within $c|x|^d$ steps on input x and outputs $D(x)$.

$\mathcal{U}$ can simulate M on input x in $c' \cdot c \cdot |x|^d \log |x|$ steps, where c' is a constant that is independent of $|x|$. **(WHY?)** There is an integer n_0 such that $n^{d+0.4} > c' \cdot c \cdot n^d \log n$ for every $n \geq n_0$. Let x be a string that encodes M such that $|x| \geq n_0$. Recall that such an x must exist, because M is encoded by infinitely many strings. Then D, on input $x = \llcorner M \lrcorner$, will compute an output bit within $|x|^{d+0.4}$ steps, but, by definition of D, that output bit will be $1 - M(x)$. Therefore, $L(M) \neq L(D)$. ∎

For nondeterministic computation, we have an even denser time hierarchy.

Nondeterministic Time-Hierarchy Theorem: If f and g are time-constructible and $f(n+1) = o(g(n))$, then $\text{NTIME}(f(n)) \subsetneq \text{NTIME}(g(n))$.

Note that these hierarchy theorems do not tell us anything about whether P is equal to NP. **(WHY?)**

In fact, we will not be able to resolve the P vs. NP problem with diagonalization, which is the technique used to prove the hierarchy theorems. This is the take-home message of the Baker-Gill-Solovay Theorem, which will be presented in class on Thursday, Feb 4, 2016.

We ended this lecture by stating the following curious fact:

Gap Theorem: There is a function $f : \mathbb{N} \longrightarrow \mathbb{N}$ such that $\text{DTIME}(f(n)) = \text{DTIME}(2^{f(n)})$.

Note that the Gap Theorem does not contradict the Deterministic Time-Hierarchy Theorm. **(WHY?)**.