Reading:

We are in Chapter 9 (on crypto) but wil transit to chapter 20 (on de-randomization) after the next lecture which is on pseudorandom generators (PRGs).

(1) Define a language A to be *downward self-reducible* if there is a polynomial-time oracle TM M such that for all n,

$$A^{=n} = L(M^{A^{<n}}).$$

That is, the status of any string of length n can be resolved by querying the status of strings of lengths $< n$. SAT is downward self-reducible because ϕ is satisfiable iff at least one of $\phi_0 = \phi[x_1 = 0]$ and $\phi_1 = \phi[x_1 = 1]$ is satisfiable—and because ϕ_0 and ϕ_1 have shorter encodings since they have one fewer variable.

Show that every such language A belongs to PSPACE. (18 pts.)

(2) Given a number $K = 2^k$, call a number m "top mod K" if $K/2 \leq (m \pmod{K}) \leq K-1$. First, find a polynomial $p_k(x, y)$ such that for all natural numbers x and y,

$$p_k(x, y) \text{ is top mod } K \iff x \text{ is top mod } K \text{ and } y \text{ is top mod } K.$$

For a hint, the famous Lucas Lemma states that a binomial coefficient $\binom{n}{m}$ is even iff for some bit i in the standard binary representations of m and n, $n_i = 0$ and $m_i = 1$.

What is the degree of your polynomial? Show that if you could construct p_k of degree $k^{O(1)}$ in time $k^{O(1)}$ then PSPACE would equal $\mathsf{P}^{\#P}$. (36 pts. total)

(3) Let $[f_n]$ be an ensemble with each f_n a function from $\{0,1\}^{2n}$ to $\{0,1\}^n$. Suppose that for each n and $w \in \{0,1\}^n$ there is a string $s_w \in \{0,1\}^n$ such that for all $x, y \in \{0,1\}^n$,

$$f_n(wx) = f_n(wy) \iff x = y \vee x = y \oplus s_w,$$

where $\oplus$ means bitwise XOR. The functions f_n are given as black boxes: if w is given and Arthur nominates a string z such that for some x, $f(wx) = z$, then Merlin can produce x and they both can see a trusted intermediary (Sir Gawain) verify that $f_n(wx) = z$. Note that w makes f_n induce a 1-to-1 function f_w from $\{0,1\}^n$ to $\{0,1\}^n$ if and only if $s_w = 0^n$.

Show that with these (somewhat artificial) settings, the problem, given w, of whether $s_w = 0^n$ belongs to AM $\cap$ co-NP. (24 pts. total)

(4) Let C be a collection of oracle Turing machines which define an ordinary class of languages C^A for any language A. We will in fact consider oracles of the form $A \cup 1R$ where R is a finite source of randomness. (The notation $0A \cup 1B$ is often written $A \oplus B$ or $A \uplus B$ and called *join* or *marked union*.) The class C by itself just means $C^{\emptyset}$. The C-machines M

are total and hence have the property that there is a computable function $r(n)$ such that for all oracles A and inputs x, $M^A(x)$ makes no query of length more than $r(|x|)$. (Proving this could be an exercise in itself—it is a consequence of *König's Lemma* applied to the tree of possible computations of $M^A(x)$ over all oracles A.) Note that $\mathcal{R}_n = \{0,1\}^{2^{r(n)+1}-1}$ thus covers all queries $M^{(\cdot)}$ can possibly make on inputs of length n.

Define "*Almost*$-\mathcal{C}$" to be the class of languages L such that for some $\mathcal{C}$-machine M (with associated $r(n)$ function, which here is a polynomial) and all n,

$$\Pr_{R \in \mathcal{R}_n}[(\forall x \in \Sigma^n) : L(x) = M^R(x)] > 3/4.$$

We can also define '*Almost*$-\mathcal{C}^A$" relative to any oracle A by making the body be $L(x) = M^{A \uplus R)}(x)$ instead. This is a finitistic way of defining the concept without needing to get into details of Lebesgue measure and "0-1 laws"; note that the "3/4" can be amplified to be as close to 1 as desired.

(a) Show that $\mathsf{BPP}[\mathcal{C}] \subseteq$ *Almost*$-\mathcal{C}$. Your proof should also work for $\mathcal{C}^A$ in place of $\mathcal{C}$ by the "general nonsense" of "relativization transparency."

(b) Sticking with the original $\mathcal{C}$ for simplicity here, can we get $\mathsf{BPP}^{\mathcal{C}} \subseteq$ *Almost*$-\mathcal{C}$? Does it suffice to assert that $\mathcal{C}$ (as an oracle class) is closed under polynomial-sized conjunctions of queries?

(c) The converse inclusions are *not* known to hold. However, they *do* hold when $\mathcal{C}$ is the relativization of NP by polynomial-time oracle NTMs. Moreover they hold when $\mathcal{C}$ is any $\sum_k^p$ or $\prod_k^p$ level of the polynomial hierarchy relativized by polynomial-time alternating TMs that make at most k alternations. This is not obvious—it uses a theorem by Nisan and Wigderson that applies to Boolean circuits of constant depth k and is touched on in Section 20.2 which we are getting into now—but we can take it as given. Deduce from this and the relativized Sipser-Gacs-Lautemann theorem $\mathsf{BPP} \subseteq \sum_2^p \cap \prod_2^p$ (which also relativizes) that if *Almost*$-\mathcal{PH}$ = *Almost*$-\sum_k^p$ for any k, then the *unrelativized*, real-world polynomial hierarchy collapses to $\sum_{k+2}^p \cap \prod_{k+2}^p$.

The standard infinitistic phrasing of theorem (by Ronald Book) in (c) is: "If the polynomial hierarchy collapses relative to a random oracle, then it collapses absolutely." This statement is now known to have a counterfactual premise: the set of oracles A such that PH^A is infinite has Lebesgue measure 1. See https://rjlipton.wordpress.com/2015/05/08/a-tighter-grip-on-circuit-depth/ if curious. But I've worded this problem so that it can be solved by "local" means within the course structure. (12+6+9 = 27 pts.)

(5) A *matrix game* (of the *2-player zero-sum* kind) has an $M \times N$ matrix A of real numbers. *Alice* secretly chooses a row i, *Bob* a row j, and after they reveal their choices, the payoff is $A[i, j]$ to Alice from Bob (so a negative entry means that Bob profits). For example, the game rock(1)-paper(2)-scissors(3) has the matrix

$$\begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$

Alice and Bob may employ *randomized strategies* which are represented by vectors α, β with nonnegative entries that sum to 1 (i.e., probability vectors). The expected value under those strategies is then

$$\alpha^T A \beta.$$

The *Minimax Theorem* of John von Neumann and Oskar Morgenstern asserts that every matrix game has a unique value v and probebility vectors α_0, β_0 (not necessarily unique) such that for all alternative strategies α', β',

$$\alpha_0^T A \beta' \geq v \qquad \text{and} \qquad \alpha'^T A \beta_0 \leq v.$$

That is, Alice has a randomized strategy α_0 that assures expectation of at least v no matter what Bob does, and Bob has a policy β_0 that asures losing no more than $-v$ per play in the long run, no matter what Alice does—even if she knows what β_0 is—as she can figure out from A given enough "pre-processing" time. In rock-paper-scissors the value is 0 (a *fair game*) and $\alpha_0 = \beta_0 = (1/3, 1/3, 1/3)$: it is in both players' best interests to play uniformly at ranodm. Note that the time for one *play* of the game can be reckoned as the number of bits in any i plus the number in any j plus the time to compute $A[i, j]$ so as to do the payoff. This allows A to have exponential size $M, N = 2^{n^k}$ for some k and still run in $O(n^k)$ time.

Now let us play the following instances of the game, given a language $L \subseteq \{0, 1\}^*$ amd a function $s(n)$ intended to bound the size of Boolean circuits according to the length of their binary string encodings (which can be reckoned as $2m \log_2 r$ where m is the number of wires and r is the number of gates). We presume that $s(n) \geq n \log_2 n$. The matrix $A_{L,s}$ has $M = 2^s = 2^{s(n)}$ rows, one for every (encoding of a) circuit C of size $s(n)$, and $N = 2^n$ columns, one for each possible input string $x \in \{0, 1\}^n$. The payoff is 1 if $C(x) = A(x)$ and -1 if not. Thus Alice chooses a size-$s(n)$ circuit and wins if it gets the correct answer on whether the string Bob chooses belongs to L.

(a) Let $v_{L,n}$ (for some fixed size function $s(n)$) stand for the value of the game at length n. Show that $v_L(n) \geq 0$ for all n.

(b) If the language L has circuits of size $s(n)$, what happens?

(c) Deduce that there is an *ensemble* $\mathcal{D} = [\mathcal{D}_n]$, each $\mathcal{D}_n$ being a probability distribution on $\{0, 1\}^n$, such that no randomized algorithm that runs in time $s(n)/\log s(n)$ can achieve more than $v_L(n)$ success per play when inputs are drawn according to $\mathcal{D}$.

(d) Show nevertheless a sense in which there is a randomized algorithm that "kind-of" runs in time $O(s(n))$ and achieves success at least a $v_L(n)$ fraction of the time, for any distribution of the inputs.

The notion of a "randomized algorithm" is rather stretched in (d), because it is not accounting for the time needed to sample *circuits* from the optimial minimax distribution computed in part (c). Modulo that, this says that the hardest distributional complexity of L equals the best possible performance of a randomized algorithm. ($6 + 6 + 9 + 9 = 30$ pts., for 135 total on the set)