

Solution Set for CPSC 468 Exam 2

Numbers of definitions, theorems, chapters, sections, etc. are from the web draft of the Arora-Barak book that was used in Fall 2007.

Question 1

Yes. There is a parsimonious reduction f from HAMPATH to SAT and another parsimonious reduction g from SAT to HAMPATH. Let h_Q be the reduction from SAT to $USAT_Q$ that proves the Valiant-Vazirani Lemma for the function Q . Then $g \circ h_Q \circ f$ is a ppt reduction from HAMPATH to $UHAMPATH_Q$.

Question 2

- (a) If $NP = P$, then $PH = P$. By the Sipser-Gacs Theorem, BPP is contained in PH; thus it is contained in P if $NP = P$. By definition, P is contained in BPP. Thus $BPP = P$ if $NP = P$.
- (b) See Definition 7.21 in Arora-Barak. Yes, UPATH is in RL; the random-walk algorithm shows that it is in RL, but, in fact, it is in $L \subseteq RL$.
- (c) L is in PP if there are a polynomial p and a deterministic polynomial-time machine M with the following properties. For all x in $\{0,1\}^n \cap L$, at least half of all w in $\{0,1\}^{p(n)}$ satisfy $M(x,w) = 1$. For all other x in $\{0,1\}^n$, strictly fewer than half of all w in $\{0,1\}^{p(n)}$ satisfy $M(x,w) = 1$. The class BPP is defined similarly except that the fraction of accepting computations of M on inputs x in L (respectively x not in L) must be of the form $\frac{1}{2} + 1/q(n)$ (respectively $\frac{1}{2} - 1/q(n)$), for some polynomial q . Because the correctness probability of a BPP machine is bounded away from $\frac{1}{2}$ by $1/\text{poly}(n)$, the BPP machine is actually a practical algorithm; application of Chernoff bounds on the tails of the binomial distribution allows us to make the error probability exponentially small with a polynomial number of independent repetitions of the BPP algorithm. This is not true of a PP machine, in which the correctness probability must be at least $\frac{1}{2}$ but need not be bounded away from $\frac{1}{2}$ by $1/\text{poly}(n)$; thus a PP machine is not a practical algorithm.

Question 3

- (a) Unknown. There is no known interactive proof system that has this property. However, if $\#P = FP$, then there is a zero-round interactive proof system for PERMANENT (i.e., the polynomial-time verifier could just compute the PERMANENT himself), but it has not (yet?) been proven that $\#P$ is not equal to FP.
- (b) False. See solution to problem 8.1.a in HW4.
- (c) Unknown. We know that, for every L in PH, there is a probabilistic polynomial-time reduction from L to parity-SAT (as in the proof of Toda's Theorem), but we do not know whether it can be made deterministic.
- (d) True. By the Sipser-Gacs Theorem, every L in BPP is in the PH and thus, by Toda's Theorem, is reducible in deterministic polynomial time to #SAT.

Question 4

- (a) $dIP = NP$ (see Theorem 8.3 in Arora-Barak), and NP is obviously contained in AM .
- (b) Let (G_1, G_2, K) be an input to this Arthur-Merlin game and m be the number of bits needed to encode a one-to-one mapping from $V(G_1)$ onto $V(G_2)$. (Note that m is approximately $n \log n$, where n is the number of vertices in each graph.) Then use the Goldwasser-Sipser lower-bound protocol from Chapter 8 of Arora-Barak to prove that the set $S \subseteq \{0,1\}^m$ of isomorphisms from G_1 to G_2 is at least K . The protocol is applicable, because membership in S can be certified; in fact, an isomorphism “certifies itself” in the sense that the verifier can check that the mapping specified by a string in $\{0,1\}^m$ is one-to-one and onto and that it preserves adjacency. The protocol can be repeated polynomially many times to increase the correctness probability.

Question 5

- (a) For every x in L , there is an oracle O such that M^O accepts x with probability 1. For every x not in L , for all oracles O^* , M^{O^*} accepts x with probability at most $1/2$. (Note that there is nothing special about $1/2$ in this definition. One would get the same class of languages if it were anything between $1/\text{poly}(|x|)$ and $1 - 1/\text{poly}(|x|)$.)
- (b) See the solution to problem 8.10 in HW5.
- (c) Let L be a language in $NEXP$ and M be a ppt oracle machine that recognizes L . For any input x , $(r, q_1, a_1, \dots, q_m, a_m)$ and $(R, Q_1, A_1, \dots, Q_m, A_m)$ are consistent transcripts if $q_j = Q_j$ implies that $a_j = A_j$. For each of the (exponentially many) distinct transcripts, there is a node in G_x , and two nodes are adjacent if and only if they represent consistent transcripts.

Question 6

- (a) Recall that a polynomial-time computable function f is a many-to-one reduction from NP language $L_1 = L(M_1)$ to NP language $L_2 = L(M_2)$ if x is in L_1 if and only if $f(x)$ is in L_2 . If f also yields a one-to-one, onto mapping between accepting computations of M_1 on x and accepting computations of M_2 on $f(x)$, then it is called parsimonious.
- (b) In the proof of the Valiant-Vazirani Lemma, a formula $\varphi(x)$ is mapped to a formula $\psi(x,y)$ that is the conjunction of $\varphi(x)$ and $\tau(x,y)$; so, if there is a unique (x,y) such that x satisfies φ and (x,y) satisfies τ , then ψ has a unique satisfying assignment. The formula τ is constructed by applying the Cook-Levin reduction to the NP language $\{(h, n, k): \text{there is an } x \text{ in } \{0,1\}^n \text{ such that } h(x) = 0^k\}$. With sufficiently high probability, the chosen h and k are such that there is a unique x satisfying $h(x) = 0^k$. (n is just the number of boolean variables in φ .) Because the Cook-Levin reduction is parsimonious, this ensures that there is a unique (x,y) that satisfies τ and hence a unique (x,y) that satisfies ψ if there are in fact any x 's that satisfy φ .
- (c) See the proof of Lemma 9.19 in Arora-Barak.

