

Solutions for Homework 4
Course: CS 509
Foundations of Computer Science

Lecturer: Joe Kilian
Department of Computer Science
Rutgers, The State University of New Jersey

November 17, 2006

Problem 1. *Suppose that (P, V) is a perfectly complete interactive proof system for L , with error $\frac{1}{4}$, where V runs in polynomial time, flipping public coins.*

A. Show that we can replace V with a verifier V' such that,

- 1. V' runs in polynomial time and logarithmic space, and throws private coins.*
- 2. $x \in L$, (P, V') will accept with probability 1.*
- 3. If $x \notin L$, (P, V') will reject with non-negligible probability (i.e., $\frac{1}{|x|^{\mathcal{O}(1)}}$).*

B. Conclude that there exists a perfectly complete interactive proof system (P', V') for L , with error $\frac{1}{4}$, where V' run in polynomial time and logarithmic space (flipping private coins).

Proof. A. In the old protocol, the verifier V has to run the computation himself. But in our new protocol, the new logarithmic space verifier V' is so lazy that he requires the prover to provide all V 's computation history and V' 's job is to randomly check the consistency of this computation transcript and behave accordingly (toss coins, halt, etc.) if it is correct, otherwise reject immediately. Therefore, if the prover gives

the real computation history, there will be no difference with respect to accepting probability; if the prover cheats, the new verifier would have $\frac{1}{|x|^{\mathcal{O}(1)}}$ chance to catch the deviation.

B. repeat the new protocol enough times to amplify the probability gap. \square

Problem 2. Recall that a language L is RP if there is a probabilistic polynomial-time decision procedure D such that

- If $x \in L$, $\Pr[D \text{ accepts } x] \geq \frac{1}{2}$, and
- If $x \notin L$, $\Pr[D \text{ accepts } x] = 0$.

A. Show that $\text{RP} \subseteq \text{NP}$.

B. Show that if $\text{NP} \subseteq \text{BPP}$, then $\text{NP} = \text{RP}$.

Proof. A. Obviously the definition of RP put itself into the class of NP .

B. The assumption means that there is a BPP algorithm for SAT , the complete problem for NP and we could make the error probability to be very small by amplification, such as $\frac{1}{2^n}$. Our solution has the same spirit as that of Problem 4 in Homework 3. So if $x \in \text{SAT}$, then with probability $1 - \frac{\text{poly}(n)}{2^n}$ (which means luckily the BPP algorithm does not make any mistake), we could find a satisfying assignment. If $x \notin \text{SAT}$, no satisfying assignments ever exist. \square

Problem 3. The solution to Problem 1 requires that the time-space bounded verifier V' use private coins. Show that if a poly-time, log-space verifier uses only public coins, then the language being accepted is in BPP (probably P suffices).

Proof. We sketch the idea of the proof here, a.k.a., dynamic programming.

Since the verifier is log-space, there are at most polynomially many possible configurations. Also the verifier runs at most polynomially many steps. Then we obtain a matrix, the rows of which denote time steps and the columns represent individual configurations. Moreover, we try to associate each entry in the matrix with a corresponding accepting probability, A . For

some entry $\langle i, \text{accept} \rangle$ with accepting configuration, we say the accepting probability $A(\langle i, \text{accept} \rangle) = 1$; for some with rejecting configuration, $A(\langle i, \text{reject} \rangle) = 0$. Others (denoted as $\langle i, p \rangle$) fall into the following three categories:

1. p is in a state to toss a coin, then it yields two possible configurations q_1, q_2 . $A(\langle i, p \rangle) = \frac{A(\langle i+1, q_1 \rangle) + A(\langle i+1, q_2 \rangle)}{2}$.
2. p is in a state to read a bit of proof, then it also yields two possible configurations q_1, q_2 , depending on the outcome of the proof bit. $A(\langle i, p \rangle) = \max\{A(\langle i+1, q_1 \rangle), A(\langle i+1, q_2 \rangle)\}$, which means currently the optimal strategy is to set the proof bit to obtain maximal accepting probability.
3. p is in regular computation and it yields q . $A(\langle i+1, p \rangle) = A(\langle i+1, q \rangle)$.

Therefore the accepting probability under optimal strategy is equal to $A(\langle 1, \text{initial} \rangle)$.

There is a small technicality remaining. We need to ensure that verifier read the proof in an online fashion (easy as an exercise, just redesign the configuration transcript). \square

Problem 4 (Sipser, 10.14). *Show that the class of nondeterministic polynomial time with a SAT oracle is equal to Σ_2^{poly} . (In Sipser's notation, show that $\text{NP}^{\text{SAT}} = \Sigma_2\text{P}$.) Note that we showed the analog to this result in the recursion theory part of this course.*

Proof. A language $L \in \text{NP}^{\text{SAT}}$ if and only if there exists a predicate P equipped with SAT oracle which nondeterministically accepts L in polynomial time.

A language $L \in \Sigma_2^{\text{poly}}$ if and only if there exist a polynomial-time predicate D and two polynomials p, q such that $\forall x \in L, \exists y, |y| = p(|x|), \forall z, |z| = q(|x|), D(x, y, z)$ accepts.

- (\leftarrow). We could nondeterministically guess a $|y|$ and ask the oracle whether $\forall z, D(x, y, z)$ accepts.
- (\rightarrow). We give the characterization as follows: $x \in L$ if and only if there are a sequence of nondeterministic choices as well as a series of guesses for

oracle queries along that branch which makes P accept x . Moreover, if we guess the oracle query to be true, there is a satisfying assignment accordingly; if we guess it false, no satisfying assignments exists.

□

Problem 5. (*Games against nature*) We say that L has a "game against nature" if there is an Arthur-Merlin proof system (P, V) for L such that

- If $x \in L$ then (P, V) accepts with probability $> \frac{1}{2}$.
- If $x \notin L$ then (P, V) accepts with probability $\leq \frac{1}{2}$ for any P .

That is, we have a proof system in which we do not have a "gap" in the accept and reject probabilities. For this problem, let us assume that V flips a fixed (polynomial) number of coins. Show that if L has a game against nature, then $L \in \text{PSPACE}$.

Proof. In PSPACE , we try to compute the success probability under the optimal strategy. Since the protocol is public coin, we will not be concerned about the consistency. The idea is to try all possible proofs and keep track of the result under current optimal strategy. Suppose for each round, the length of V 's coin tosses is $p(n)$ and the length of P 's proof is $q(n)$. Basically the following claim suffices.

Claim 0.0.1. *If the communication is at most i rounds, we could compute the success probability under the optimal strategy using space at most $cip(n)q(n)$, where c is a constant.*

Proof of claim. When $i = 1$, this is easy. For each coin toss, we try all proofs to look for one which makes the verifier accept and call this coin toss "good". Therefore, the success probability under the optimal strategy is just the portion of "good" coin tosses.

Assume that the claim is true for $i = k - 1$. When $i = k$, we do the similar things as the base case. For each initial coin toss, we try all proofs to look for one which makes the verifier accept with maximal probability under optimal strategy and under the assumption, we are able to do that within $c(k - 1)p(n)q(n)$ space because the remaining is a $k - 1$ round game. Then summing these probability up and dividing it by $\frac{1}{2^{p(n)}}$, we obtain our goal in the statement. □

Since the game only has polynomially many rounds, the claim implies that $L \in \text{PSPACE}$. □