

# CS 509: Take Home Final

Due December 15

Rules of the game: This is to be your work and your work only. You may not consult any other person regarding the questions in this exam. You may of course request clarification on these questions; all clarifications will be posted on the web site.

This exam is open book. However, you should not seek out specific solutions to these questions on the web, in textbook teacher's editions, etc.

**Problem 1:** For a (possibly probabilistic) Turing machine  $M$  and a function  $f$ , let the  $f$ -truncation of  $M$ , denoted  $M_{[f]}(x)$  be defined operationally as follows: Run  $M(x)$ , but if  $M(x)$  tries to flip more than  $f(|x|)$  coins, halt and reject.

- A. Suppose that  $M$  runs in expected polynomial time and that for some  $L$ ,  $M(x)$  decides  $x \in L$  with error  $\leq 1/10$ .<sup>1</sup> Show that for some polynomial  $f$ ,  $M_{[f]}(x)$  decides  $x \in L$  with error  $\leq 1/4$ .
- B. Give an example in which a probabilistic  $M$  runs in polynomial space, decides  $x \in L$  with error  $\leq 1/10$ , yet for any polynomial  $f$ , for some  $x$ ,  $M_{[f]}(x)$  is wrong with probability 1.
- C. Repeat Part B, except that  $f$  can be any exponential function.

**Problem 2:** In class, we defined the complexity classes PPP1 and PPP2 as follows. A language  $L$  is in PPP1 iff there exists a probabilistic Turing machine  $M$  such that  $M$  runs in constant expected time (after which it must halt and either accept or reject), and the following holds:

$$\begin{aligned}x \in L &\rightarrow \Pr(M \text{ accepts on input } x) > \frac{1}{2} \\x \notin L &\rightarrow \Pr(M \text{ accepts on input } x) \leq \frac{1}{2}\end{aligned}$$

A language  $L$  is in PPP2 iff there exists a probabilistic Turing machine  $M$  such that  $M$  runs in constant expected time (after which it must halt and either accept or reject), and the following holds:

$$\begin{aligned}x \in L &\rightarrow \Pr(M \text{ accepts on input } x) > \frac{1}{2} \\x \notin L &\rightarrow \Pr(M \text{ accepts on input } x) < \frac{1}{2}\end{aligned}$$

Note the subtle difference. In class, I mentioned that PPP1 corresponds to the class of recursively enumerable languages and PPP2 corresponds to the class of recursive (decidable) languages. For this problem, we establish this result (note that one of them was done for you in class):

---

<sup>1</sup>That is, if  $x \in L$ ,  $M(x)$  accepts with probability  $\geq .9$  and if  $x \notin L$ ,  $M(x)$  accepts with probability  $\leq .1$ .

- A. Show that if  $L$  is recursively enumerable, then  $L \in \text{PPP1}$ .
- B. Show that if  $L \in \text{PPP1}$ , then  $L$  is recursively enumerable.
- C. Show that if  $L$  is decidable, then  $L \in \text{PPP2}$ .
- D. Show that if  $L \in \text{PPP2}$ , then  $L$  is decidable.

**Hint:** For any  $M$ , consider the  $a(M, x, i)$ , the probability that  $M$  accepts  $x$  within  $i$  steps,  $r(M, x, i)$  the probability that  $M$  rejects  $x$  (halts without accepting) within  $i$  steps, and  $h(M, x, i)$  the probability that  $M$  halts on  $x$  within  $i$  steps. Consider how these quantities can be computed, how they change as  $i$  gets larger and larger, and when their values imply that  $x \in L$ .

**Problem 3:** Define the mystery class  $Q$  of languages by  $L \in Q$  iff there is a probabilistic polynomial-time machine  $M$  such that

$$\begin{aligned} x \in L &\rightarrow \Pr(M \text{ accepts on input } x) = 1 \\ x \notin L &\rightarrow \Pr(M \text{ accepts on input } x) \leq \frac{1}{2} \end{aligned}$$

I may have given this definition as the one for RP - if so, I was wrong! However, it is equal to a class we did study, or its complement.

- A. What is  $Q$ ?
- B. Prove it.

**Problem 4:** Let us consider what happens with interactive proofs in which the verifier is a probabilistic polynomial time Turing machine with access to an oracle for SAT. That is, it can solve any NP-complete problem “for free.” We call these verifiers *augmented verifiers*. We know that the set of languages accepted by interactive proof systems with ordinary verifiers is equal to PSPACE. What about augmented verifiers?

- A. Show that if language  $L$  is accepted by a one-sided (perfectly complete) public coin interactive proof system (AKA an Arthur-Merlin game) with an augmented verifier, then  $L \in \text{PSPACE}$ .

**Hint:** Show that we can convert the proof system to one that works with ordinary verifiers.

- B. (\*) Show that even if  $L$  is accepted by a private coin interactive proof system with an augmented verifier, then  $L \in \text{PSPACE}$ .

**Hint:** What transformation can we apply?

**Problem 5:** [Note, I changed the clauses from the original - may still need to be tuned...]

Given a CNF formula  $F$ , let  $\text{maxfrac}(F)$  denote the maximum number of clauses of  $F$  that one can satisfy, divided by the number of clauses in  $F$ . One way of stating the PCP theorem is that for some fixed constant  $c < 1$ , it is NP-hard to distinguish whether  $\text{maxfrac}(F) = 1$  or

$\text{maxfrac}(F) \leq c$ , where  $F$  is a 3CNF formula. What if we restrict ourselves to 2-CNF formulas (conjunctions of disjunctions of at most 2 literals). Let us show that in this case, there are fixed constants  $c_1 < c_2$  such that it is NP-hard to distinguish whether  $\text{maxfrac}(F) \geq c_2$  or  $\text{maxfrac}(F) \leq c_1$ . Let  $F'$  be a 3CNF formula over  $x_1, \dots, x_n$ :

$$F' = (t_{1,1} \vee t_{1,2} \vee t_{1,3}) \wedge \dots \wedge (t_{m,1} \vee t_{m,2} \vee t_{m,3}),$$

where each literal  $t_{i,j}$  corresponds to either a variable  $x_k$  or its negation,  $\overline{x_k}$ . We construct a 2CNF formula  $F$  as follows.  $F$  contains all the variables in  $F'$  and  $3m$  auxiliary variables,

$$y_{1,1}, \dots, y_{m,3}.$$

For each clause  $C'_i = (t_{i,1} \vee t_{i,2} \vee t_{i,3}) \in F'$ ,  $F$  contains the 9 clauses

$$\begin{aligned} (y_{i,1} \rightarrow t_{i,1}) \quad (y_{i,2} \rightarrow t_{i,2}) \quad (y_{i,3} \rightarrow t_{i,3}) \\ (y_{i,1} \rightarrow \overline{y_{i,2}}) \quad (\overline{y_{i,1}} \rightarrow y_{i,2}) \\ (y_{i,1} \rightarrow \overline{y_{i,3}}) \quad (\overline{y_{i,1}} \rightarrow y_{i,3}) \\ (y_{i,2} \rightarrow \overline{y_{i,3}}) \quad (\overline{y_{i,2}} \rightarrow y_{i,3}) \end{aligned}$$

where  $(a \rightarrow b)$  is syntactic sugar for  $(\overline{a} \vee b)$ .

- A. Show that any setting of the variables must violate at least 1 of the above 9 clauses.
- B. Show that if an assignment to  $x_1, \dots, x_n$  satisfies  $C'_i$ , then there exist a setting of  $y_{i,1}, y_{i,2}, y_{i,3}$  such that 8 of the above 9 clauses will be satisfied.
- C. Show that if an assignment to  $x_1, \dots, x_n$  fails to satisfy  $C'_i$ , then for any setting of  $y_{i,1}, y_{i,2}, y_{i,3}$ , at most 7 of the above clauses will be satisfied.
- D. Bring it home. Show that there exist constants  $c_1 < c_2$  such that

$$\begin{aligned} \text{maxfrac}(F') = 1 &\rightarrow \text{maxfrac}(F) \geq c_2, \text{ and} \\ \text{maxfrac}(F') \leq c &\rightarrow \text{maxfrac}(F) \leq c_1, \end{aligned}$$

Where  $c$  is as in the PCP theorem.