Honors Theory of Computation/Spring 2000/Chee Yap/March 22
MidTerm with Solution

ANSWER ALL QUESTIONS. THIS IS AN OPEN BOOK, IN-CLASS EXAM. PLEASE WRITE ONLY IN COMPLETE ENGLISH SENTENCES (EVEN IF THEY INVOLVE CENTERED EQUATIONS). UP TO 10% OF THE POINTS ARE FOR THE CLARITY OF YOUR PROOFS.

Problem 1 (Short Questions) [10 points for each part]

No credit will be given without some short justification. Be sure to distinguish between FALSE and UNKNOWN. Even in the UNKNOWN case, you must also give some facts that are relevant to the statement. E.g., if you say that $\mathrm{TIME}(f) \subseteq \mathrm{NSPACE}(g)$ is unknown, you would probably cite the known results about when a TIME-class is included in a NSPACE-class, and note why they are not enough to show the desired inclusion.

(a) True/False/Unknown: Let REC and RE denote (respectively) the class of recursive and the class of recursively enumerable languages. Then $\mathrm{REC} = \mathrm{RE} \cap \text{co-}\mathrm{RE}$.

SOLUTION True. This is a simple but well-known result. Writing the justification may take a bit longer:

First we show that $\mathrm{REC} \subseteq \mathrm{RE} \cap \text{co-}\mathrm{RE}$. If $L \in \mathrm{REC}$ then it is decided by some TM M that halts on all inputs. Clearly $L \in \mathrm{RE}$. The complement of L is accepted by a TM M' where M' is the same M except that the notion of acceptance and rejection is interchanged. This proves that $L \in \text{co-}\mathrm{RE}$. Conversely, if $L \in \mathrm{RE} \cap \text{co-}\mathrm{RE}$, there are two TM's M and M' where M accepts x iff $x \in L$ and M' rejects iff $x \notin L$. We now dovetail the computation of M and M', and take the answer of whichever machine that gives us an answer. This dovetail process clearly decides for each x whether x is in L.

COMMENT: you could have just quoted a result stated in the text instead of giving this argument. Many students who offered an argument, only prove ONE of the two directions. You loose 3 points for this.

(b) True/False/Unknown: $\mathrm{NTIME}(n) \subseteq \mathrm{TIME}(2^n)$.

SOLUTION Unknown. We know that $\mathrm{NTIME}(n) \subseteq \cup_{k>1} \mathrm{TIME}(k^n)$, but we do not know how to fix k to any constant ("$k = 2$" in the particular question here).

COMMENT: Some of you cite theorem 7.4 (p.147) in Papadimitrious' book. There are two problems: it does not give the strongest version of known inclusion between NTIME and TIME (you should use theorem 2.6 (p.47) instead). The second problem is that theorem 7.4, part (c), is incorrectly stated: the correct statement is $\mathrm{NSPACE}(f(n) \subseteq \cup_{k>1} \mathrm{TIME}(k^{\log n + f(n)})$

If you are aware of theorem 2.6, you should haved realized that the answer is UNKNOWN. However, many students fall into a subtle error. You thought you could reduce all nondeterministic machines to just 2 choices, and therefore the answer is TRUE. Unfortunely, this transformation of nondeterministic machines causes the running time to increase from $f(n)$ to $c \cdot f(n)$ for some constant c. When you simulate this machine deterministically, you will need $2^{cf(n)}$ time, which is NOT $O(2^{f(n)}$.

(c) True/False/Unknown: $\text{co-}\mathrm{NSPACE}(2^n) \subseteq \mathrm{SPACE}(5^n)$.

SOLUTION True. By the result of Immerman and Szelepscényi, we know that $\text{co-}\mathrm{NSPACE}(2^n) = \mathrm{NSPACE}(2^n)$. By Savitch's theorem, we know $\mathrm{NSPACE}(2^n) \subseteq \mathrm{SPACE}(4^n) \subseteq \mathrm{SPACE}(5^n)$.

(d) True/False/Unknown: $\mathrm{NSPACE}(\sqrt{\log n}) \subseteq \mathrm{SPACE}(\log n)$.

SOLUTION Unknown. The closest relevant result we know is Savitch's theorem which tells us when $\mathrm{NSPACE}(f)$ is contained in $\mathrm{SPACE}(g)$, but only if $f(n) \geq \log n$.

(e) True/False/Unknown: SPACE(n) $\subset$ SPACE($n \log\log n$). Note that "$\subset$" denotes proper subset.

SOLUTION True. This is just a straightforward application of the Space Hierarchy Theorem proved in homework 3 (but not from the corresponding theorem in the text).

(f) True/False/Unknown: TIME(n) $\subset$ TIME($n \log\log n$).

SOLUTION Unknown. The Time Hierarchy Theorem says that if $f(n)\log n = o(g)$ then $dtime(f) \subset \text{TIME}(g)$. But this is not strong enough to be applied to this situation.

(g) True/False: The number 3 is a quadratic residue modulo 23.

SOLUTION True: Compute the Legendre symbol (3/23). $(3/23) = -(23/3) = -(2/3) = +1$.

A slightly more tedious computation (but easily doable by hand) is to use Euler's Criterion: $(a/p) \equiv a^{(p-1)/2} (\bmod\, p)$ if p is prime. Thus we check that 3^{11} **mod** $23) = 1$.

COMMENT: if you find the answer by enumeration (e.g., $7^2 \equiv 3 (\bmod\, 23)$, I take off one point).

(h) True/False: The number 23 has fewer than 5 primitive roots.

SOLUTION False. In fact, we proved in class (and in the text) that p has $\phi(p-1)$ primitive roots. Hence, 23 has $\phi(\phi(23)) = \phi(22) = 10$ primitive roots.

Problem 2 (Completeness) [30 points]

Let us say that a language L is **cubically reduced** to L' if there is some function f that can be computed in time $O(n^3)$ such that $x \in L$ iff $f(x) \in L'$. In notation, write $L \leq_3 L'$ (via f). Using this notion of reduction, we can define (as usual) the notion of "complete language" for a class K of languages: a language L_0 is **cubically complete** for K if $L_0 \in K$ and, for all $L \in K$, $L \leq_3 L_0$. Prove that the class P of polynomial-time decidable languages has no cubically complete languages.

HINT: If L_0 is cubically complete for P then L_0 is decided in time n^k for some k. How hard is it to decide L if $L \leq_3 L_0$?

SOLUTION: Using the hint, suppose L_0 is decided in time n^k for some $k \geq 1$. Then that L can be accepted in time n^{3k}: to see this, assume that L is cubically reduced to L_0 via f. On input x, we decide if $x \in L$ as follows: (1) compute $f(x)$ and (2) decide if $f(x) \in L_0$. The first step takes time $O(n^3)$ and the second takes time $O((n^3)^k)$. Thus L can be decided in time $O(n^{3k})$. Thus $P \subseteq \text{TIME}(n^{3k})$. According to the Time Hierarchy Theorem, there is a language in TIME(n^{9k}) that is not in TIME(n^{3k}), contradiction!

COMMENT: Some students assert contradiction as follows (without invoking the Hierarchy Theorem): They say that L is decided in time n^c for some c. Since L can be solved in $O(n^{3k})$, this implies $c \leq 3k$. This is a contradiction because c can be any integer greater than $3k$. What is wrong with this argument?

Another common mistake is to say that if $L \leq_3 L_0$ then L can be decided in time $O(n^3 + n^k)$.

Problem 3 (RP) [30 points]

In chapter 9, there is a characterization of languages in NP in terms of polynomial-decidable and polynomial-balanced languages. Give a similar characterization for languages in RP.

SOLUTION: Say that a binary predicate $R \subseteq \Sigma^* \times \{0,1\}^*$ is "randomly-good" if there is some $k \geq 1$ such that for all $x \in \Sigma^*$, for all $y \in \{0,1\}^*$, if $R(x,y)$ holds, then (1) $|y| \leq |x|^k$, and (2) $R(x,z)$ holds for more than half of the strings z of length $\leq |x|^k$.

Our characterization of RP is this: L is in RP iff there is a polynomial-decidable and randomly-good predicate $R \subseteq \Sigma^* \times \{0,1\}^*$ such that L is the **projection** of R, i.e., $L = \{x : (\exists z)R(x,z) holds\}$.

(⇒) Let M be an RP-machine for L, and we may assume that for all inputs of length n, every path in the computation of M halts in exactly n^k steps for some k. Also, assume M makes only binary choices in its computation path. We just let $R(x, y)$ be the predicate which says that $|y| = |x|^k$ and the path corresponding to y leads to an acceptance by M on input x. Clearly R is polynomial-decidable. It is also randomly-good because M is an RP-machine.

(⇐) Let L be the projection of $R(x, y)$, a polynomial-decidable predicate that is randomly-good. Assume that if $R(x, y)$ holds, then $|y| = |x|^\ell$, and assume that R is decided by a TM M_R in time exactly n^k. Without loss of generality, assume $\ell \leq k$ (since we can artificially make M_R run in time n^ℓ in case $\ell > k$). We define another predicate R' as follows: $R'(x, y)$ holds iff (i) $|y| \leq |x|^k$ and (ii) for some prefix y' of y, $R(x, y')$ holds. Clearly, if R is randomly-good then R' is randomly-good. This construction shows that we might as well assume that $k = \ell$.

We now show that L in in RP. On input x, we first guess (non-deterministically) a string y of length $|x|^k$. Then we run the machine M_R to decide if $R(x, y)$ holds. We claim that this is an RP-procedure which accepts x iff $x \in L$. Since M_R is deterministic and runs for n^k steps, we can view its computation as a complete binary tree of depth n^k in wich every leaf accepts or every leaf rejects. If $x \in L$ then more than half of the guessed strings y leads M_R to accept, which means more than $(n^k/2)(n^k) = n^{2k}/2$ of the leaves accept. But there are n^{2k} leaves in the computation tree. Conversely, if $x \notin L$ then no leaves accept. Thus L is in RP.