MidTerm + Solution

This is a closed book, one-hour in-class exam. Please write ONLY in complete English sentences (even if they involve centered equations). The quality of your proofs is also being graded.

Problem 1 (R.E. sets) [20 points]
If true, prove it; if false, show a counter example. You may freely use facts from the textbook, but please state them first.
(a) T/F: Suppose a language A is r.e.. Then its complement co-A is also r.e.
(b) T/F: If the three languages B, $C \setminus B$ and $B \setminus C$ are r.e., then C is r.e.. (NOTE that $C \setminus B$ denotes set difference.)

SOLUTION 1
(a) False. Let $A = A_{TM}$ (p.160, Sipset). Then A is r.e.. If co-A is also r.e., then A would be recursive, which is a contradiction.
(b) False. Let $B = \Sigma^*$ and $C =$co-A_{TM}. Then B, $C \setminus B = \emptyset$ and $B \setminus C = A_{TM}$ are r.e. However, we know that C is not r.e..

Problem 2 (CFL) [25 points]
In this problem, you will be strongly graded on your ability to give a "textbook quality proof": be as logical and concise as possible, yet be explicit about your assumptions. Show that the language

$$L = \{1^{n^2} : n \in \mathbb{N}\} \subseteq \{1\}^*$$

is not context-free.

SOLUTION 2 For the sake of clarity, I will write the proof in bulletted items.

- We will use a proof by contradiction: suppose L is context-free.
- Then by the Pumping Lemma for context-free languages, there is an n such that for all $w \in S$, if $|w| \geq n$ then w can be written in the form $w = uvxyz$, $|vy| > 0$ and for all $i \geq 0$, $uv^ixy^iz \in S$.
- We can clearly choose a word $w \in S$ such that $|w| \geq n$ and the above property holds. For every $i \geq 0$, there exists an m such that $|uv^ixy^iz| = m^2$. Choose i large enough so that $m > |vy|$.
- Then $|uv^{i+1}xy^{i+1}z| = m^2 + |vy| < m(m+1) < (m+1)^2$. Hence $m^2 + |vy|$ is not a perfect square, contradiction.

Problem 3 (Recursion Theorem) [15 points]
Use the recursion theorem to show the existence of a Turing machine M that, on any input x, erases the tape and writes the string

$$\langle \mathrm{M} \rangle^R$$

on the tape and halts. Note that $\langle \mathrm{M} \rangle$ is the standard encoding for M, and for any string w, w^R denotes the reversal of w.
NOTE: you are required to explicitly specify the function $t(\cdot)$ used in the statement of the recursion theorem.

SOLUTION 3 We construct the function $t : \Gamma^* \to \Gamma^*$ used in the recursion theorem. The function t is computed by a Turing machine T which behaves as follows: "on input of the form $u\#v$ where u is #-free, T replaces the input with the string u^R". By the recursion theorem, there is a TM M that computes the function $r(v) = t(\langle M\rangle\#v)$ for all v. This machine M, on any input v will simulate the computation of T on $\langle M\rangle\#v$, which simply prints the string $\langle M\rangle^R$ on the tape. Thus M has the desired behavior.

Problem 4 (Complexity) [30 points]
Let $s(n)$ be a total complexity function such that $\log_2 n \leq s(n) < \infty$ for all n. Prove that the running complexity class

$$DSPACE_r(s(n))$$

is closed under complement. HINT: you need to detect whether a machine is looping using $s(n)$ space.

SOLUTION 4 Suppose M is a multitape TM that runs in space $s(n)$. It is sufficient to construct another multitape TM N such that N simulates M with with following properties:

- N uses no more space than M.
- If M loops on an input x, then N detects this and rejects.
- In all other cases, N behaves just like M.

We now make two other assumptions:

- By the tape reduction result in the notes, we assume M has only one work tape.
- We may assume that the number of distinct configurations of M, on inputs of length n, is at most 2^{kn} for some integer $k \geq 1$. Note that k depends on the number of states and symbols in M.

N will also have one work tape, but this work tape will have two tracks: the upper track will simulate the work tape of M. The lower track will be a 2^k-ary counter whose space usage never exceed the space used in the upper track. On any input x, N simulates the behaviour of M, but for each step of M, N will increment the counter on the lower track. If N ever reaches the limit of the counter (i.e., the counter attempts to use more space than the upper track), it means M has looped, and N will reject the input. Otherwise N simulates M to completion.