CMP 132 Handout #6
David Helmbold
February 6, 2003

First Midterm Review

This homework set is a review for the midterm, and is generally not to be turned in.

Thus far we have covered chapters 9, 10, and 11.1-11.3 in the 3rd edition (9, 10, 11, and 12.1-12.3 in the 2nd edition). Any of the material in those chapters may be on the midterm. The topics include:

1. Turing Machines, transition tables, transition diagrams, configurations, accepting languages, Church-Turing thesis.
2. Computing functions with Turing Machines, how to handle multiple arguments.
3. Combining Turing Machines and standard subroutines (insert, delete, copy, concatenate, addition, etc.).
4. Variations on Turing Machines: Multiple tapes better? Two way tapes better? Need to write Δ? Is a finite tape OK?
5. Non-deterministic Turing Machines and how to "simulate" them with deterministic Turing Machines.
6. Universal Turing Machines, "standard" encoding of Turing Machines, letters, and strings.
7. The Recursive and Recursively Enumerable languages, and their basic definitions (L accepted by TM, TM computes characteristic function for L).
8. Enumeration characterization of Recursive and Recursively Enumerable languages.
9. Properties of Recursive Languages: closed under union, intersection, complement, concatenation. "Time-splicing" and the "triangle trick".
10. Properties of Recursively Enumerable Languages: closed under union, intersection, concatenation.
11. Countably infinite versus uncountable sets.
12. Not all languages Recursively Enumerable – counting argument and diagonalization.
13. Languages NSA and SA.
14. Unrestricted Grammars and how they can be "programmed"
15. Proofs that the languages generated by unrestricted grammars are exactly the Recursively Enumerable languages.
16. Context Sensitive grammars and linear bounded automata.
17. Chomsky Hierarchy, and some recursive languages are not context sensitive.
18. Decision problems and Languages
19. Reductions and reduction functions
20. Tricks for reductions: "brain surgery" on TM's to add pre-processing, postprocessing, crash-freeness, or replacing h_r by an infinite loop.

Here are some kinds of possible questions:

1. Given a TM figure out what it does (the language accepted or function computed).
2. Construct a TM for solving a simple problem.
3. What is the definition of a term on from the above? What are simple consequences of the definition?
4. Prove a simple result from the above list (like the closure properties of Recursive or Recursively Enumerable languages, Theorems 10.1–10.6 in the book, the equivalence between enumeration by TM and acceptance by TM, or one of the diagonalization arguments).
5. Something about countability and uncountability.
6. Give an unrestricted grammar generating a certain language, or figure out the language generated by a certain unrestricted grammar.
7. A problem (or part of a problem) from the homeworks.
8. One of the simple reductions in the text (like showing that **halting** reduces to **Acepts(Δ)**.

Some study problems from 2nd edition:

1. The extra "recommended" problems on homeworks 3 and 4
2. 9.3, 9.8, 9.11, 9.13,
3. 10.1, 10.9, 10.12 (recall that finite sets are countable), 10.17, 10.19 (b),
4. 11.3 (a) (d), 11.10, 11.12, 11.22
5. 12.1, 12.2, 12.3, 12.11

Very Optional Extra Credit Problem (may be turned in before exam, probably very hard, studying for the midterm is more important):

> Prove that every context sensitive language (as defined by definition 11.3 (2nd) or 10.4 (3rd)) has a grammar where every production has the form $\alpha A\beta \rightarrow \alpha X\beta$ where A is a variable and α, β and X are strings of variables and/or terminals.