7 Homework

Due: Thursday, March 14, 2013.

Instructions

- Please, review the homework grading policy outlined in the course information page.
- On the *first page* of your solution write-up you *must* make explicit which problems are to be graded for regular credit, which problems are to be graded for extra credit, and which problems you did not attempt. Use a table that looks like this:

Problem	1	2	3	4	5	6	7	8	9	...
Credit	RC	RC	RC	EC	RC	EC	NA	NA	EC	...

  where "RC" denotes "regular credit", "EC" denotes "extra credit", and "NA" denotes "not attempted". Failure to include such a table will result in an arbitrary set of problems being graded for regular credit, no problems being graded for extra credit, and a 5% penalty assessment.
- You must also write down with whom you worked on the assignment. If this varies from problem to problem, write down this information separately with each problem.

Problems

Required: 5 of the following 6 problems
Points: 20 points per problem

1. Do Problem 3.4

   Give a formal definition of an *enumerator*. Consider it to be a two-tape Turing machine that uses its second tape as the printer. Include a definition of the enumerated language.

2. Do Problem 3.6 and 3.7

   Problem 3.6

In Theorem 3.21 in the text, we showed that a language is Turing-recognizable iff some enumerator enumerates it. Why didn't we use the following simpler algorithm for the forward direction of the proof? As before, s_1, s_2,.... is a list of all strings in Σ^*.

$E =$ "Ignore the input.

1. Repeat the steps 2. and 3. for $i = 1, 2, 3$ ...

2. Run M on s_i.

3. If it acceptes, print out s_i."

Problem 3.7

Explain why the following is not a description of a legitimate Turing machine.

$M_{bad} =$ "On input $< p >$, a polynomial over variables x_1, x_2, ... x_k:

1. Try all possible settings of x_1, x_2, ... x_k to integer values.

2. Evaluate p on all these settings.

3. If any of these settings evaluates to 0, accept, otherwise reject."

3. Do Problems 3.8 parts b and c

   Give implementation-level description of Turing machines that decide the following languages over the alphabet $\{0, 1\}$.

   (b) $\{w | w$ contains twice as many 0s as 1s$\}$.

   (c) $\{w | w$ does not contain twice as many 0s as 1s$\}$.

4. Do Problem 3.13

   A **Turing machine with stay put instead of left** is similar to an ordinary Turing machine, but the transition function has the form

   $\delta : Q \times \Gamma \to Q \times \Gamma \times \{R, S\}$.

   At each point, the machine can move its head right or let it stay in the same position. Show that this Turing machine variant is *not* equivalent to the usual version. What class of languages do these machines recognize?

5. Do Problems 3.15 part b and 3.16 part b

   Problem 3.15 (b)

   Show that the collection of decidable languages is closed under concatenation.

   Problem 3.16 (b)

   Show that the collection of Turing-recognizable languages is closed under concatenation.

6. Do Problems 3.15 part e and 3.16 part d

   Problem 3.15 (e)

   Show that the collection of decidable languages is closed under intersection.

   Problem 3.16 (d)

   Show that the collection of Turing-recognizable languages is closed under intersection.