COMS W3261: Theoretical Computer Science.
Instructor: Tal Malkin

Problem Set 8

Due: Tue, 11/13/07
No late homework will be accepted.
Reading: Chapter 3

Before you start: Read "Terminology for describing Turing Machines", starting on page 156 in the text. This will guide you as to the level of detail you need to provide in describing Turing Machines in your solutions (a formal description, implementation level description, or high-level description.) For all three, you should add whatever explanations are necessary to make sure that it is clear how your machine works (e.g., explain what different states are supposed to capture, etc).

Also, I recommend you go over some examples in the text (try to solve them yourself before checking their solution) to get more experience with Turing Machines. In particular, check examples 3.11, 3.12, 3.23 (these examples are not related to the questions below, but may be generally helpful in getting the hang of Turning Machines).

1. (a) (10 points) Construct a Turing machine which decides the language $L = \{w | w$ contains the string ab exactly once$\}$ over the alphabet $\Sigma = \{a, b\}$.

   Give a detailed formal description of the Turing Machine. The transition function of your TM can be described either with a table, or with a state diagram.

   (b) (4 points) Provide the sequence of configurations of your TM when started on each of the following four inputs: $abab, aaba, \epsilon, bb$.

2. (10 points) An input-output Turing Machine (IO-TM) is defined similarly to a TM, except it has only one halting state, q_{halt} (replacing q_{acc} and q_{rej}). The IO-TM T computes the function (or partial function) f if for any input x on which f is defined, when the machine starts from the start configuration on input x, the machine halts with $f(x)$ written on its tape, followed by blanks.

   Give implementation-level description for an input-output Turing Machine which computes the function $f(x) = 2x$, where x is represented in unary. That is, on input a^n, the output is a^{2n}.

3. (16 points) 3.14 in text (proving that a queue automaton is equivalent to a TM). You may use the result of 3.12 (namely that a TM with left reset is equivalent to a standard TM) or any other variants that we proved equivalent in class, if that helps you.

4. (a) (10 points) Prove that the class of Turing-recognizable languages is closed under the concatenation operation.

   (b) (10 points) Prove that the class of Turing-decidable languages is closed under the complementation operation.

   (c) (8 points) Does your proof of part (a) work to prove the same claim for the class of Turing-decidable languages? Does your proof of part (b) work to prove the same claim for the class of Turing-recognizable languages? Explain your answers.