CSU390 Theory of Computation	Assignment #06
Spring 2006	March 3, 2006

Homework 06

Due: Friday, March 17, 2006

Note: This assignment cannot be accepted late because solutions will be distributed at the March 17 class meeting in preparation for Exam 2.

Instructions

1. Please review the homework grading policy outlined in the course information page.
2. 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.
3. 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 7 problems
Points: 20 points per problem

1. Give both an informal implementation-level description and a state transition diagram for a Turing Machine that decides the language $\{w \mid w \in \{\mathtt{a}, \mathtt{b}\}^*$ and w is a palindrome$\}$.
2. Another way to deal with the problem of finding the beginning of a TM tape besides those mentioned in the text is to insert a special symbol (say, \$) at the beginning of the tape and slide the entire input string one cell to the right. Give both an informal implementation-level description and a state transition diagram for a TM (operating as a transducer) that does this. In addition, it should reposition the tape head at the start of of the actual input string, one cell to the right of the \$. To keep it simple, assume that the alphabet is $\{\mathtt{a}, \mathtt{b}\}$, but explain how your solution would change if the alphabet had more symbols in it.
3. Give both an informal implementation-level description and a state transition diagram for a Turing Machine transducer that accepts as input a string in $\{\mathtt{a}, \mathtt{b}\}^*$ and counts the number of `a`'s in binary. The binary representation of the number of `a`'s on the tape should appear immediately following the input string, and this binary representation should appear in reverse order (i.e., with least significant bit first). Some examples: if there are 4 `a`'s, the result should be 001 (followed by blanks); if there are 13 `a`'s, the result should be 1101; if there are no `a`'s, the result should be 0. *Hint:* This can be done by making a small change to the example shown in class that counts the number of `a`'s in unary.
4. • Prove that the collection of decidable languages is closed under complementation and intersection.

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

For this problem, give only informal high-level descriptions of any required Turing Machines. *Hint:* You may find it helpful to use nondeterministic and/or multi-tape Turing Machines.

5. - Prove that the collection of decidable languages is closed under concatenation and star.
   - Prove that the collection of Turing-recognizable languages is closed under concatenation and star.

   For this problem, give only informal high-level descriptions of any required Turing Machines. *Hint:* You may find it helpful to use nondeterministic and/or multi-tape Turing Machines.

6. Define a k-PDA to be a pushdown automaton that has k stacks, where $k \geq 0$. Thus a 0-PDA is an NFA and a 1-PDA is a conventional PDA.
   - Show that any k-PDA, for given $k \geq 0$, can be simulated by a Turing Machine. *Hint:* You may find it helpful to use a nondeterministic multi-tape TM for this.
   - Show that any Turing Machine can be simulated by a 2-PDA.

7. Given an arbitrary Turing machine (or Turing machine variant) M, let M' be the same machine but with the accept and reject states swapped. Is it possible that there exist strings accepted by:

   i. both M and M'; or

   ii. neither M nor M';

   when:

   a. M is a (deterministic) decider?

   b. M is a (deterministic) recognizer?

   c. M is a nondeterministic decider?

   d. M is a nondeterministic recognizer?

   Note that 8 answers are required. Justify all answers.