Harvard University
Computer Science 121
Problem Set 5

Due Friday, October 29, 2010 at 1:20 PM.
Submit a single PDF (lastname+ps5.pdf) of your solutions to cs121+ps5@seas.harvard.edu
Late problem sets may be turned in until Monday, November 1, 2010 at 1:20 PM with a 20% penalty.
See syllabus for collaboration policy.

Name

Problem set by !!! Your Name Here !!!

with collaborators !!! Collaborators' names here !!!!

Notes

Try this week's BONUS problem! We think you'll like it.
Use the provided template for your answers. Custom formatted solutions are difficult to grade.
Read instructions carefully. Don't forger one side of an IFF.

PROBLEM 1 (4 + 4 + 2 + 2 points)

Let $G = (V, \Sigma, R, S)$ where $V = \{S, V\}$, $\Sigma = \{a, b\}$, and R is the set of rules:

$$S \to bSS \mid aS \mid aV$$
$$V \to aVb \mid bVa \mid VV \mid \varepsilon$$

(A) Transform G into an equivalent grammar G' in Chomsky normal form.

(B) Verify that the string $abaab$ is generated by G', using the recognition algorithm for grammars in Chomsky normal form given in class. Show the complete filled-in matrix.

(C) Draw a parse tree for the derviation of $abaab$ from the transformed grammar G'.

(D) In one sentence, what language does G generate?

PROBLEM 2 (6 points)

A left-linear rule of a CFG (V, Σ, R, S) is one whose right-hand side is a member of $V\Sigma^* \cup \Sigma^*$, that is, only the leftmost symbol can be a nonterminal. Right-linear rules are defined analogously. Show that a grammar with only left-linear rules, or only right-linear rules, generates a regular language, but a grammar with a mixture of left- and right-linear rules may generate a non-regular language.

PROBLEM 3 (6 points)

Consider two languages, L_1 and L_2, recognized by Turing machines M_1 and M_2 respectively. Prove there is a Turing machine, M_{1+2} which recognizes $L_{1+2} = L_1 \cap L_2$. You may use the multitape Turing Machine model.

PROBLEM 4 (6 + 2 points)

Consider a Turing machine M with a tape alphabet of size $n \geq 2$ deciding a language L.

(A) Show how to construct a Turing machine with a two-symbol tape alphabet that decides L.

(B) Compare your constructed machine's and M's time to halt. That is, if M takes N steps to halt on input w, approximately how many steps, as a function of N, n, and perhaps other parameters of M, will your transformed machine take?

PROBLEM 5 (6 + 2 points)

A queue is similar to a stack, except that pushing and popping happen at opposite ends. That is, symbols are pushed onto the right of the queue, and symbols are popped off the left of the queue.

A queue automaton (QA) is a deterministic automaton that, in addition to having a finite set of states and transitions, has a queue for data storage. At each step, a QA dequeues the next symbol, and based on that symbol and its current state, transitions to another state and queues any number of symbols . When run on an input string w, a QA begins with the string $w\$$ in the queue, where \$ is a symbol not in the input alphabet that marks the end of the string. A QA accepts by transitioning to a special accept state. A QA can fail to accept by looping forever or if its queue becomes empty (so that it cannot dequeue a symbol and make any more transitions).

(A) Demonstrate how a Turing machine can be simulated by a QA. You need not prove your construction correct, but it should handle corner cases correctly.

(B) Estimate the number of states in your construction of a QA as a function of the number of states in the TM and the size of its alphabet.

PROBLEM 6 (+1 BONUS points)

Prove the language of palindromes, $L = \{ww^R \mid w \in \Sigma^*\}$ is not recognized by any deterministic pushdown automata.