Harvard University
Computer Science 121
Problem Set 3

Due Friday, October 8, 2010 at 1:20 PM.
Submit a single PDF (lastname+ps3.pdf) of your solutions to cs121+ps3@seas.harvard.edu
Late problem sets may be turned in until Monday, October 11, 2010 at 1:20 PM with a 20% penalty.
See syllabus for collaboration policy.

Name

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

with collaborator !!! Collaborators' names here !!!!

Notes

When drawing your parse tree you may want to draw by hand or use a graphics program and import like we did for ps1.
When not stated elsewhere assume the alphabet $\Sigma = \{a, b\}$.
Use the provided template for your answers. Custom formatted solutions are difficult to grade.
Read instructions carefully. DRAW means exactly that.
We denote the reverse (defined naturally) of a string w to be w^R

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

Which of the following languages are regular? Support each of your answers with a proof.

(A) $\{a^i b^j a^j b^i \colon\ i,\ j \geq 0\}$.
(B) $\{a^{i^2} \colon\ i \geq 0\}$.
(C) $\{wxw^R : w, x \in \Sigma^*\}$
(D) $\{ww^R\}$.

PROBLEM 2 (4+1 points)

(A) Prove the following strong form of the pumping lemma: there is a number p where, if s is any string in L of length at least p, then s may be written as $s = xyz$ such that

1. for each $i \geq 0$, $xy^i z \in L$,
2. $|y| > 0$, and
3. $|xy| \leq p$.

(B) The pumping lemma states every regular language L has these properties. Is the converse true? That is, if a language has these properties is it regular? Describe why or why not briefly.

PROBLEM 3 (4 points)

Show that a DFA with n states accepts an infinite language if and only if it accepts some string of length at least n and less than $2n$.

PROBLEM 4 (2+2 points)

(A) Prove the set of finite languages over a fixed alphabet is countable.

(B) Prove that if A is an uncountable set and B is a countable set, then $A - B$ is uncountable.

PROBLEM 5 (4+2+2 points)

A satisfied boolean expression is a string over the alphabet $\Sigma = \{0, 1, (,), \neg, \wedge, \vee\}$ representing a boolean expression that evaluates to 1, where $\neg$ is the *not* operator, $\wedge$ is the *and* operator, and $\vee$ is the *or* operator (see Sipser p.14). For instance, 1, $(1 \wedge (\neg 0))$, and $(\neg(\neg 1))$ are satisfied boolean expressions.

The following are examples of strings that are *not* satisfied boolean expressions:

$)(1$
$(0\neg)$
$\wedge(0 \vee 1)$
$(0 \wedge 1)$
$(\neg(1 \vee 0))$

The first three examples are strings that aren't valid expressions and simply don't make sense. The last two are well-formed expressions, but they evaluate to 0, rather than to 1.

(A) Write a context-free grammar that generates the language of satisfied boolean expressions. You may assume that expressions must be completely parenthesized. For instance, your grammar need not generate $\neg(0 \wedge \neg 0 \wedge 1)$ but should generate its equivalent, $(\neg(0 \wedge ((\neg 0) \wedge 1)))$.

(B) In several sentences, explain why the language of satisfied boolean expressions is not regular.

(C) Draw the parse tree in your grammar for the expression $((\neg(1 \wedge 0)) \vee (\neg(0)))$.

PROBLEM 6 (4+4+2 points)

A context-free grammar G is **ambiguous** if there exists a string $w \in L(G)$ with two distinct leftmost derivations in G. (That is, two distinct derivations in which the leftmost nonterminal is replaced at each step).

(A) Show that the context-free grammar $G = (V, \Sigma, R, S)$, where $V = \{S, T\}$, $\Sigma = \{a, b\}$, and

$$R = \{S \to aT,\ T \to S,\ T \to Tab,\ ,\ T \to a\}$$

is ambiguous.

(B) A language L is **inherently ambiguous** if all context-free grammars G such that $L = L(G)$ are ambiguous. Show that if L is a regular language, then L is **not** inherently ambiguous. (*Hint*: think about regular grammars).

(C) Show that $L(G)$, where G is the context-free grammar in Part (A), is **not** inherently ambiguous.

PROBLEM 7 (Challenge!! 1 points)

Show that for any subset L of $\{a\}^*$, the language L^* is regular.