

## Homework 4

Due midnight, Thursday, 2/19/2015

Please submit your homework solutions online at <http://www.dci.pomona.edu/tools-bin/cs081upload.php>. If you have more than one file to be turned in, please put it in a folder and zip it up before turning it in.

Problems from the texts are given in the form c.n where c is the chapter and n is the problem number. Thus problem 2.7 is problem 7 from Chapter 2.

1. (10 points) **Chomsky Normal Form**

We consider the strings of balanced parentheses of two types over the alphabet is  $\Sigma = \{ (, ), [, ] \}$ . The language  $\text{PAREN}_2$  is the smallest set of strings satisfying the following three properties.

- (a)  $\epsilon \in \text{PAREN}_2$ ;
- (b) if  $x$  is in  $\text{PAREN}_2$ , then so are  $(x)$  and  $[x]$ ; and
- (c) if  $x$  and  $y$  are in  $\text{PAREN}_2$ , then so is  $xy$ .

Give a context-free grammar in Chomsky normal form for  $\text{PAREN}_2 - \{\epsilon\}$ .

2. (10 points) **Ambiguity and Parse Trees**

Rich 11.10

3. (10 points) **CFL to PDA**

Use the construction in the text or lecture to create a pda that accepts the language generated by the grammar for arithmetic expressions with  $V = \{\text{Exp}, \text{Addop}, \text{Term}, \text{Factor}, \text{Mulop}, \text{num}, *, +, -, /, (, )\}$ ,  $\Sigma = \{\text{num}, *, +, -, /, (, )\}$ , start symbol Exp, and the following rules:

- $\text{Exp} \rightarrow \text{Exp Addop Term} \mid \text{Term}$
- $\text{Term} \rightarrow \text{Term Mulop Factor} \mid \text{Factor}$
- $\text{Factor} \rightarrow (\text{Exp}) \mid \text{num}$
- $\text{Addop} \rightarrow + \mid -$
- $\text{Mulop} \rightarrow * \mid /$

*Be sure to use the given construction. We will see later that it is undecidable for two pda's  $M$  and  $M'$  whether  $L(M) = L(M')$ . The TA's get irritated if they have to solve undecidable problems in order to grade the homework!!*

4. (20 points) **PDA Variants**

Let  $M = (K, \Sigma, \Gamma, \Delta, s, A)$  be a pda. The language accepted by M by final state is defined as follows:

$$L_f(M) = \{w \in \Sigma^* \mid (s, w, \epsilon) \vdash_M^* (f, \epsilon, \alpha) \text{ for some } f \in A, \alpha \in \Gamma^*\}$$

This differs from the definition in Rich by allowing acceptance even when the stack is non-empty.

Note that to show two languages  $L_1$  and  $L_2$  are the same (e.g., as in the two parts below), it is generally simplest to show that  $L_1 \subseteq L_2$  and the reverse. You will need to do this in each of the cases below to show the languages are the same!

- (a) For every pda  $M$ , show there is a pda  $M'$  s.t.  $L(M') = L_f(M)$ . Notice that one uses the subscript  $f$  and the other does not.

*Hint: Let  $M'$  be a variant of  $M$  that allows the possibility of emptying its stack whenever  $M$  enters a final state.*

- (b) For every pda  $M$ , show there is a pda  $M'$  s.t.  $L_f(M') = L(M)$ .

*Hint: Let  $M'$  begin by putting a special marker on top of the stack that will allow a check to see if the stack is otherwise empty.*

5. (5 points) **Non-context-free Languages**

Rich 13.1.i. *Hint: You will need to look at cases. See the examples in the book.*

6. (10 points) **Closure**

Rich 13.13.ef.

7. (15 points) **Closure**

Rich 13.22bcd. *Be careful, these are tricky!*