5DV037 Fundamentals of Computer Science Fall 2010
Obligatory Exercise 3
Due date: October 11, 2010 at 8am (0800)

1. Let $G = (V, \Sigma, S, P)$ be the context-free grammar defined by

$$\begin{aligned}
V &= \{S, A, B, C, D, E\} \\
\Sigma &= \{a, b, c\}. \\
P = \{ \quad & S \to AB \mid ABC \\
& A \to aAb \mid bAa \mid a \mid \lambda \\
& B \to bba \mid AAC \mid CBB \mid b \mid \lambda \\
& C \to ccC \mid ccDC \\
& D \to DDDb \mid bbbC \mid AB \\
& E \to S \mid ab \qquad\qquad \}
\end{aligned}$$

Using the algorithm given in the textbook or in the course slides, construct an equivalent reduced context-free grammar (*i.e.*, with no useless productions). To obtain credit, you must show your work.

2. Let $G = (V, \Sigma, S, P)$ be the context-free grammar defined by

$$\begin{aligned}
V &= \{S, A, B, C\} \\
\Sigma &= \{a, b, c\}. \\
P = \{ \quad & S \to AB \mid ABC \\
& A \to BA \mid BCC \mid a \mid \lambda \\
& B \to AAC \mid CBB \mid b \mid \lambda \\
& C \to BC \mid ABC \mid AAA \mid c\}
\end{aligned}$$

Using the algorithm given in the textbook or in the course slides, construct an equivalent context-free grammar which is nonerasing and without chain rules. To obtain credit, you must show your work.

3. Design an NPDA which accepts the language

$$L = \{\alpha \in \{a, b\}^* \mid \mathsf{Count}\langle a, \alpha\rangle \leq \mathsf{Count}\langle b, \alpha\rangle\}$$

by final state. Express your solution as a state-transition diagram. To obtain credit, you must explain how your solution works.

5DV037, Obligatory Exercise 3, page 2

4. Let $G = (V, \Sigma, S, P)$ be the context-free grammar defined by

$$\begin{aligned} V &= \{S, A, B, C\} \\ \Sigma &= \{a, b\}. \\ P &= \{ \quad \begin{array}{l} S \to C \\ A \to BC \mid a \\ B \to CA \mid b \\ C \to AB \qquad \} \end{array} \end{aligned}$$

(a) Using the method described in the course slides, construct a basic top-down NPDA parser which accepts $\mathcal{L}(G)$ by final state. Express your solution as a state-transition diagram.

(b) Draw a parse tree for the string *bababa*.

(c) Show the sequence of moves which your answer to (a) goes through in accepting the string *bababa*.

Note: The construction of Theorem 7.1 of the textbook for constructing a parser for a CFG, while similar to the one found in the course slides, requires that the grammar be in Greibach normal form. Thus, it will not work on this grammar.

5. Repeat Problem 4, this time constructing a one-state basic top-down NPDA parser which accepts $\mathcal{L}(G)$ by empty store. (You do not have to give the parse tree for *bababa* again, of course.)

Note: Th construction for part (a) is not found in the textbook, but it is spelled out clearly in the course slides.

Further Notes:

1. As stipulated in the course syllabus, this exercise may be done either individually, in a group of two, or in a group of three. Remember that there are point penalties for late submission. See the course syllabus.
2. It is strongly recommended that you use a graphical tool to display your results. If you draw them by hand, they must be very neat. It is not allowed to copy the work of others. The submission must be the original work of the individual(s) in the working group. The grader reserves the right to interview members of the working group about the solution.
3. So that solutions may be discussed in a class meeting, students and/or groups that are very late in preparing a solution may be required to solve an alternate problem to receive credit for this exercise.