Tutorial 5,6

1. Suppose one has two stacks instead of one stack in NPDA. Intuitively, the NPDA can now push (possibly different) strings on the two stacks, and base its actions on the top symbol of each of the stacks as well as on the input symbol.

   Formally define a two stack NPDA. Is it more powerful than one stack NPDA (that is can it accept something which cannot be accepted by one stack NPDA)?

2. A DPDA (or deterministic push down automata) is just like an NPDA, but all its moves are deterministic, that is, in each state, for each top of stack symbol and input symbol, there is at most one possible next move. Additionally, if there is an ϵ move for some state q and top of stack symbol A, then there is no move involving state= q, top of stack = A, and any input symbol in Σ.

   (a) Formally define DPDA.

   (b) Can all regular languages be accepted by a DPDA?

3. Suppose a NPDA (that accepts by final state) never pushes more than one symbol. That is, for all p, q, a, Z, if $(p, \gamma) \in \delta(q, a, Z)$, then $|\gamma| \leq 1$. Then, show that the language accepted by the NPDA is regular.

   For the following, the lower case letters are members of the alphabet Σ, and the upper case letters are non-terminals.

4. Remove useless symbols from the following grammar using the algorithm done in class.

$$S \to A|AA|AAA$$
$$A \to ABa|ACa|a$$
$$B \to ABa|Ab|\epsilon$$
$$C \to Cab|CC$$
$$D \to CD|Cd|CEa$$
$$E \to b$$

5. Eliminate ϵ productions from the grammar

$$S \to ABaC$$
$$A \to AB$$
$$B \to b|\epsilon$$
$$C \to D|\epsilon$$
$$D \to d$$

6. Remove all unit productions from the grammar

$$S \to CBa|D$$
$$A \to bbC$$
$$B \to Sc|ddd$$
$$C \to eA|f|C$$
$$D \to E|SABC$$
$$E \to gh$$

7. Convert the following to Chomsky normal form grammar without useless symbols:

$$S \to AB|CA$$
$$A \to a$$
$$B \to BC|AB$$
$$C \to aB|b|ACC|\epsilon$$

8. Give an algorithm to test whether the language generated by a CFG is (a) empty, (b) finite, (c) infinite?

9. Assume G is a grammar without any ϵ productions. Let $Unit(A) = \{B : A \Rightarrow_G^* B\}$. Give an algorithm that constructs $Unit(A)$ for all nonterminals A in G.

10. (Hard) Greibach Normal Form: A grammar is said to be in Greibach Normal Form, if all the productions in the grammar are of the form: $A \to a\alpha$, where a is a terminal and α is a string of zero or more terminals/non-terminals. Prove that, for every non-empty context free language L not containing ϵ, there is a Greibach Normal Form grammar.

    Hint: Assume the original grammar given for the language L is in Chomsky Normal form. Assume that the non-terminals in the grammar are $A_1, \ldots, A_m$. Let G_0 be the original grammar.

    (a) First, inductively define G_i (generating the same language L) to have the following properties:

    (P1) G_i has non-terminals $A_1, \ldots, A_m$ and $B_1, \ldots, B_i$,

    (P2) all the productions of G_i are of form (i) $A_j \to \alpha$ (where α starts with either a terminal, or a variable A_r, with $r \geq \min(i+1, j+1)$), OR (ii) $B_j \to \alpha$, where α starts with a terminal or A_k for some k.

    The above can be achieved as follows. Suppose in G_{i-1} we have productions of the form $A_i \to \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_r$ and $A_i \to A_i\beta_1 \mid A_i\beta_2 \mid \ldots \mid A_i\beta_w$, where α_s either start with a terminal or A_k for some $k > i$ (note that by inductive property P2, above holds). Now replace the above productions by:

    $A_i \to \alpha_1 B_i \mid \alpha_2 B_i \mid \ldots \mid \alpha_r B_i$

    $A_i \to \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_r$

    and

$B_i \to \beta_1 \mid \beta_2 \mid \ldots \mid \beta_w$,

$B_i \to \beta_1 B_i \mid \beta_2 B_i \mid \ldots \mid \beta_w B_i$

Here if β_r starts with a $B_{r'}$, $r' < i$, then replace $B_{r'}$ in these productions by the RHS of all productions of $B_{r'}$.

(note that above is "correct" replacement as the language generated does not change).

Now for $j > i$, replace each production in G_{i-1} of form $A_j \to A_i\gamma$ by the set of productions $A_j \to \alpha_1 B_i\gamma \mid \alpha_2 B_i\gamma \ldots \mid \alpha_r B_i\gamma$ and

$A_j \to \alpha_1\gamma \mid \alpha_2\gamma \ldots \mid \alpha_r\gamma$.

Now verify that the grammar so generated, G_i, satisfies the properties (P1) and (P2).

(b) Let us rename the non-terminals in G_m as

B_i renamed to C_i

A_j renamed to C_{m+j}.

Then, we have the property that any production of form $C_r \to \alpha$, has α starting with either a terminal or a variable C_w, where $w > r$. Use this property to convert the grammar into Greibach normal form.