Tutorial 6

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

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

   Ans: The generating symbols (besides the terminals a, b, d) are:

   A, B, E in first induction step and then S in the next induction step.

   Non-generating symbols are C and D.

   Dropping them gives the grammar:

   $S \to A|AA|AAA$

   $A \to ABa|a$

   $B \to ABa|Ab|\epsilon$

   $E \to b$

   Now, in the above grammar, the nonterminals S, A, B are reachable, and E is non-reachable. Dropping E gives the grammar:

   $S \to A|AA|AAA$

   $A \to ABa|a$

   $B \to ABa|Ab|\epsilon$

2. Eliminate ϵ productions from the grammar

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

   Ans:

   B, C are nullable.

   Thus, the new grammar will be:

   $S \to ABaC|AaC|ABa|Aa$

$A \to DB|D$

$B \to b$

$C \to D$

$D \to d$

3. Remove all unit productions from the grammar

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

Ans: Unit pairs (besides the trivial ones) are: $(S, D), (D, E), (S, E)$. (where $(S, D), (D, E)$ are obtained in the first induction step, and (S, E) is obtained in the next induction step).

Thus, the new productions are:

$S \to CBa|SABC|gh$

$A \to bbC$

$B \to Sc|ddd$

$C \to eA|f$

$D \to SABC|gh$

$E \to gh$

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

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

Ans: B is a useless (non-generating) symbol. Thus, eliminating it gives:

$S \to CA$

$A \to a$

$C \to b|ACC|\epsilon$

All symbols in above grammar are reachable.

C is nullable, thus eliminating ϵ productions gives:

$S \to CA|A$

$A \to a$

$C \to b|ACC|AC|A$

The unit pairs (besides the trivial ones) are: $(S, A), (C, A)$. Thus, eliminating unit productions gives:

$S \to CA|a$

$A \to a$

$C \to b|ACC|AC|a$

All the symbols in the above grammar are useful (generating and reachable).

Converting now to Chomsky Normal Form we get:

$S \to CA|a$

$A \to a$

$C \to b|AZ|AC|a$

$Z \to CC$

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

   Ans:

   (a) Check if S is generating or not (algorithm is done in class). If S is generating, then CFG generates a non-empty language. Otherwise it generates an empty language.

   (b,c):

   (i) Convert the grammar to Chomsky normal form without useless symbols. Note that this at most removes ϵ from the language, and thus does not effect its finiteness.

   (ii) Using the CNF grammar without useless symbols construct a graph, using the new grammar as follows:

   — Vertices of the graph are the non-terminals in the grammar.

   — There is a directed edge from A to B iff there is a production in the grammar with A on the left hand side and B on the right hand side.

   (iii) If there is no directed cycle in the graph, then clearly the language generated is finite: Any parse tree for the elements of the language has depth bounded by the number of nonterminals in the grammar. Thus, there are only finitely many possible parse trees.

   If there is a directed cycle in the graph, then the language generated is infinite. To see this, suppose A is part of a loop. Then there is derivation as $S \Rightarrow^* \gamma A\delta \Rightarrow^* \gamma\alpha A\beta\delta$, where $\gamma, \delta, \alpha, \beta$ consist only of terminals. Furthermore, at least one of α, β is non-empty (as the grammar didn't have any ϵ productions or unit productions or useless symbols).

   Suppose $A \Rightarrow^* w$ (as A is generating).

   Now, the language contains $\gamma\alpha^k w\beta^k\delta$ for all k. Thus, it contains arbitrarily large strings and thus infinite.

6. 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.

   Ans:

1. Let $UP = \{(A, A) : A \in V\}$
2. Repeat:
   3. Done=True
   4. For each pair $(A, B) \in UP$ and each non-terminal $C \in V$ Do:
      If $B \to C$ is a production and $(A, C) \notin UP$, do:
      $UP = UP \cup \{(A, C)\}$
      Done=False
      Endif
      Endfor
   Until Done

   At the end UP gives the unit pairs.

   Now $Unit(A) = \{(B : (A, B) \in UP\}$.

7. (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, which does not contain ϵ, one can have a Greibach Normal Form grammar.

   Ans (sketch):

   Below, upper case letters in the productions stand for non-terminals.

   Assume the original grammar given for the language L is in Chomsky Normal form without useless symbols. Assume that the non-terminals in the grammar are $A_1, \ldots, A_m$. Let G_0 be the original grammar.

   (a) We 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 the form (i) $A_j \to \alpha$ (where α starts with either a terminal, or a non-terminal 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. Note that, all productions in G_{n-1}, except for productions of form $A_r \to A_n\beta$, for $r \geq n$, satisfy the requirements (P1) and (P2) for G_n. We will see how to modify these productions to convert G_{n-1} to G_n.

   First suppose the productions in G_{n-1} for A_n are:

   $A_n \to \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_r$ and

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

where α_s either start with a terminal or A_k for some $k > n$ (note that by inductive property P2, above holds) and β_s are non-empty, without loss of generality.

Now replace the above productions by:

$A_n \to \alpha_1 B_n \mid \alpha_2 B_n \mid \ldots \mid \alpha_r B_n$

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

and

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

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

Here if β_j starts with a $B_{j'}$, $j' < n$, then replace $B_{j'}$ in these productions by the RHS of all productions of $B_{j'}$ (in G_{n-1}).

Note that above is "correct" replacement as the language generated does not change.

Now for $j > n$, replace each production in G_{n-1} of the form $A_j \to A_n\gamma$ by the set of productions

$A_j \to \alpha_1 B_n\gamma \mid \alpha_2 B_n\gamma \ldots \mid \alpha_r B_n\gamma$ and

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

To see that G_n satisfies property (P1) note that only non-terminal B_n is introduced in the above modification.

To see that G_n satisfies property (P2), note that $A_j \to \ldots$ already satisfy the property by induction for $j < n$. Similary for B_j, with $j < n$.

For the new productions which were introduced above for $A_n \to \ldots$, note that α_r either started with a terminal or with A_k, $k > n$. So these satisfy $P2$. Similarly, for RHS of $A_j \to \ldots$, with $j \geq n$ either RHS started in G_{n-1} with a terminal or a nonterminal A_k, with $k > n$ or with $k = n$. If it started with A_n, then we updated it to start with α_r which started with terminal or a $A_{k'}$ with $k' > n$.

Similarly, we made sure that the right hand side of any production with LHS being B_n started with a terminal or some A_k. Thus, P2 is satisfied.

(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 the form $C_r \to \alpha$, has α starting with either a terminal or a non-terminal C_w, where $w > r$.

Now we can use a grammar with the above property to convert it into Greibach normal form as follows.

Let the above grammar be denoted by H_{2m}.

Inductively, construct grammar H_i, starting with $i = 2m - 1$ down to $i = 1$ as follows: replace each production of the form $C_i \to A\alpha$ by the productions $\{C_i \to \beta\alpha : A \to \beta$ in $H_{i+1}\}$.

Now H_1 is in Greibach normal form.