CS3230
Tutorial 9

1. Compute the longest common subsequence of

   (a) $SLWOVNNDK$ and $ALWGQVNBBK$.

   Ans: $LWVNK$

   (b) $AGCGATAGC$ and $ACAGATGAG$

   Ans: $ACGATAG$

2. Show that if X and Y are two sequences starting with a, then the longest common subsequence of X and Y starts with an a.

   Ans: Consider any longest common subsequence of X and Y, say $s_1 s_2 \ldots s_k$. Then, for some $1 \leq r_1 < r_2 < \ldots < r_k \leq |X|$ and for some $1 \leq r'_1 < r'_2 < \ldots < r'_k \leq |Y|$, $s_i = X[r_i] = Y[r'_i]$. If $s_1 \neq a$, then $r_1 > 1$ and $r'_1 > 1$. But then $as_1 s_2 \ldots s_k$ is also a common subsequence of X and Y which is of length greater than k. A contradiction.

3. Give a counterexample to the following claims:

   (a) If $X = X[1] \ldots X[m]$ and $Y = Y[1] \ldots Y[n]$, and $X[m] \neq Y[n]$, then the longest common subsequence of X and Y must end in either $X[m]$ or $Y[n]$.

   Ans: Let $X = abac$ and $Y = abab$.

   (b) If $X = X[1] \ldots X[m]$ and $Y = Y[1] \ldots Y[n]$, and $X[m] \neq Y[n]$, then the longest common subsequence of X and Y must not end with either $X[m]$ or $Y[n]$.

   Ans: Let $X = abbc$ and $Y = abab$.

4. Consider the following problem:

   A relation on a set A is a subset of $A \times A$.

   A relation T is called transitive if the following holds for all $a, b, c \in A$:

   $$\text{if } (a, b) \in T \text{ and } (b, c) \in T, \text{ then } (a, c) \in T.$$

   A relation T is called a transitive closure of a relation R on a set A if it is the smallest relation (on A) which is a superset of R and is transitive. In other words, (a, b) is in T iff there exist $b_1, b_2, \ldots, b_k$ such that, $a = b_1, b = b_k$, and $(b_1, b_2), (b_2, b_3), (b_3, b_4), \ldots, (b_{k-1}, b_k)$ are all in R (here k may be equal to 2).

   Give a dynamic programming algorithm to compute transitive closure T of a relation R, given the relation R as a matrix.

Ans: Suppose B is the matrix giving the relation R over the set $A = \{1, 2, \ldots, n\}$.

Then, consider the following algorithm:

For $k = 1$ to n {
    For $i = 1$ to n {
        For $j = 1$ to n {
            $A[i, j] = A[i, j]$ or $(A[i, k]$ and $A[k, j])$.
        }
    }
}

5. (a) A Chomsky Normal Form grammar G is of the form $G = (\Sigma, V, S, \delta)$, where Σ is the alphabet set, V is a set of non-termimals (where $V \cap \Sigma = \emptyset$), $S \in V$ is a starting symbol and δ is a set of productions of the form:

$$A \to BC \text{ or } A \to a, \text{ where}$$

$A, B, C \in V$ and $a \in \Sigma$ is a terminal.

In the following α, β, γ, w are strings in $(\Sigma \cup V)^*$.

(b) We say that $\alpha A \beta \Rightarrow_G \alpha w \beta$, where $A \to w$ is a production in G (that is member of δ).

(c) We say that $\alpha \Rightarrow_G^* \beta$ if one of the following holds: (i) $\alpha = \beta$, (ii) $\alpha \Rightarrow_G \beta$ or (iii) for some γ, $\alpha \Rightarrow_G \gamma$ and $\gamma \Rightarrow_G^* \beta$.

(d) We say that $L(G) = \{w : w \in \Sigma^* \text{ and } S \Rightarrow_G^* w\}$.

Given a Chomsky Normal Form grammar G, give a dynamic programming algorithm to determine if a string $w \in \Sigma^*$ is a member of $L(G)$.

Ans: Suppose $w = w_1 w_2 \ldots w_n$.

For $1 \leq i \leq j \leq n$, let $X_{i,j}$, be the set of nonterminals A such that $A \Rightarrow_G^* w_i w_{i+1} \ldots w_j$.

Then, $A \in X_{i,i}$, iff $A \to w_i$ is a production in the grammar.

For $1 \leq i < j \leq n$, $A \in X_{i,j}$, iff $A \to BC$ and $B \in X_{i,k}, C \in X_{k+1,j}$, for some k such that $i \leq k < j$.

Then, finally $w \in L(G)$ iff $S \in X_{1,n}$.

To compute $X_{i,j}$, one can compute them in order of increasing difference $j - i$.

1. For $i = 1$ to n, let $A \in X_{i,i}$, iff $A \to w_i$ is a production.
2. For $r = 1$ to $n - 1$ {
    For $i = 1$ to $n - r$ {
    Let $j = i + r$.
    Let $X_{i,j} = \emptyset$.
    For $k = i$ to $j - 1$ {
        For each production $A \to BC$ in the grammar do, {

If $B \in X_{i,k}$ and $C \in X_{k+1,j}$ then let $X_{i,j} = X_{i,j} \cup \{A\}$.
}
}
}
}