CS3231
Tutorial 11

1. Show that the following problem is undecidable:

   Given two CFGs G_1 and G_2, is $L(G_1) \subseteq L(G_2)$?

   Ans: Let G_A and G_B as given in the lecture.

   Note that $\overline{L(G_A)}$ and $\overline{L(G_B)}$ are also context free.

   Define G_1 and G_2 such that:

   $L(G_2) = \overline{L(G_A)} \cup \overline{L(G_B)} = \overline{L(G_A) \cap L(G_B)}$

   and $L(G_1) = (\Sigma \cup I)^*$.

   Note that there exist such CFG G_1 and G_2, and G_2 can be found algorithmically from G_A and G_B.

   We have that $L(G_1) \subseteq L(G_2)$ iff $L(G_A) \cap L(G_B) = \emptyset$. Thus, using undecidability of PCP, we have that given arbitrary grammars G_1 and G_2, whether $L(G_1) \subseteq L(G_2)$ is undecidable.

2. Show that the following problem is undecidable: Given two CFGs G_1 and G_2, is $L(G_1) = L(G_2)$?

   Ans: Same G_1, G_2 as for Q1 works here too.

3. Give unrestricted grammar for

   (a) $\{a^i b^j c^k \mid i \leq j \leq k\}$.

   Ans:

   The method is similar to the one done in class for the language $\{a^i b^i c^i : i \geq 1\}$.

   Below R is used to generate strings in the language which do not contain a, and T is used to generate strings which contain at least one a.

   $S \to R \mid T$

   $R \to bRc \mid Rc \mid \epsilon$

   $T \to aTBC \mid TBC \mid TC \mid \epsilon$

   $CB \to BC$

   $aB \to ab$

   $bB \to bb$

   $bC \to bc$

   $cC \to cc$

   (b) $\{w \mid w \in \{a, b, c\}^*$ and w contains equal number of a's, b's and c's$\}$.

   Ans:

   This is also similar to the grammar done in class for $\{a^i b^i c^i : i \geq 1\}$, except that this time we allow A, B, C to move around in arbitrary order.

$S \to SABC \mid \epsilon$

$AB \to BA$

$BA \to AB$

$AC \to CA$

$CA \to AC$

$CB \to BC$

$BC \to CB$

$A \to a$

$B \to b$

$C \to c$

(c) $\{ww \mid w \in \{a, b\}^*\}$.

Ans:

In the following grammar, initially one generates pairs of aA and bB ending with an X. Then, A, B are shifted towards the right end over a, b (just before the X).

The second last and third last rules are then used to push the A, B to the right of X in form of a, b respectively. Last rule eliminates the X.

$S \to aAS \mid bBS \mid X$

$Aa \to aA$

$Ab \to bA$

$Ba \to aB$

$Bb \to bB$

$AX \to Xa$

$BX \to Xb$

$X \to \epsilon$

4. Yes for both. Suppose S_1, S_2 are the starting symbols for two grammars for two context sensitive languages (whose non-terminals do not intersect). We also assume without loss of generality that productions do not have terminals on the left hand side. This is without loss of generality as we can have a dummy non-terminal A for terminal a, and then have a production of the form $A \to a$. Note that we need a different "A" for the two grammars as we are assuming non-terminals do not intersect.

   Now, consider grammar which has the productions of both the above grammars as well as:

   $S_u \to S_1 \mid S_2$,

   with S_u being the starting symbol, then this grammar would generate the union of the two languages.

   If one considers

$S_c \to S_1 S_2$,

with S_c being the starting symbol, then this grammar would generate the concatenation of the two languages.

5. Yes, context sensitive languages are closed under intersection.

   Context sensitive languages are exactly the languages which can be accepted in linear space by a non-deterministic Turing Machine. Given two Turing machines M_1, M_2 which are non-deterministic, we can construct another non-deterministic Turing Machine which can accept the intersection by running the two non-deterministic machines in parallel (in different set of tapes, after copying the input) and accepting iff both the machines accept the input.

6. (a) True. Let B be regular language $\{1\}$ over the alphabet $\{0, 1\}$. Now, for any context free language A, let $f(x) = 1$, if $x \in A$ and $f(x) = 0$ if $x \notin A$. f can be computed in polynomial time using the CYK algorithm done in class.

   (b) True. If the graph G is a complete graph, then it is not $n-1$ colorable. Otherwise, pick (u, v) such that $u \neq v$ and $(u, v) \notin E$. Let $f(u) = f(v) = 1$, and color the other $n-2$ vertices using colors 2 to $n-1$. Thus, the graph is colorable using $n-1$ colors iff it is not a complete graph.