CS 381 – HW5 Solutions

1. Given any CF grammar $G = (\Sigma, N, S, P)$, construct a grammar G' such that $L(G') = \{reverse(w) \mid w \in L(G)\}$.

Solution 1:

Define $G' = (\Sigma, N, S, P')$ where the only difference between G and G' are the production rules. We define the new production rules $P' = \{reverse(p) \mid p \in P\}$. The notation $reverse(p)$ simply means we reverse the production rule p. So if $p = AxBy$ then $reverse(p) = yBxA$.

2. Construct a PDA M so that $L(M) = \{0^l 1^k \mid k \leq l \leq 2k\}$.

Solution 2:

The PDA M will push one marker onto the stack for every 0 it reads at the beginning of the input word. The PDA will then non-deterministically pop one or two of these symbols off the stack for every 1 it reads at the end of the input word. Clearly, the number of markers pushed onto the stack will be exactly l where the input word began 0^l. The number of symbols that are popped off the stack can be any number inclusively between k and $2k$ where k is the number of 1s in the string. We accept as long as this number is exactly l (the symbols pushed onto the stack), which means we accept if and only if $k \leq l \leq 2k$, as desired.

3. Prove that $\{w \; reverse(w) \; w \mid w \in \{0,1\}^*\}$ is not a CFL.

Solution 3:

We can appeal to the pumping lemma for CFLs (more demon fun!). The demon gives us n, we give the demon the word $x = 0^n 1 0^2 n 1 0^2 n 1 0^n$ which is in our language because $x = w \; reverse(w) \; w$ for $w = 0^n 1 0^n$. The demon partitions $x = x_1 x_2 x_3 x_4 x_5$ where $|x_2 x_3 x_4| \leq n$ and $|x_2 x_4| \geq 1$. Lets switch on the demon's choice of $x' = x_2 x_3 x_4$. Because $|x'| \leq n$ and each 1 in string x is surrounded by n or more 0s, there is at most one 1 in x'. Supposing there is a 1 in $x_2 x_4$, then we can pump zero times to delete the 1 to produce a string with only two 1s. This is clearly not in the desired language because in any word in our language all symbols appear in multiples of three (once in the first w, once in the $reverse(w)$, and finally in the last w). If the 1 does not appear in $x_2 x_4$ (either it is in x_3 or not in x' at all), pumping changes the length of either one sequence of 0s or two sequences of 0s. However, there are four sequences of 0s in our original word, and changing the length of two sequences must remove the word from the language. The word $0^i 1 0^j 1 0^k 1 0^l$ is in our language only if $i + l = j = k$, and pumping one or two 0 sequences cannot leave a word of this form.

4. Prove that if L_1 is a CFL and L_2 is a regular language, then $L_1 \cap L_2$ is a CFL.

Solution 4:

We can prove this by constructing a PDA that accepts the intersection $L_1 \cap L_2$. Lets take the states from the PDA for L_1 and the states from the DFA for L_2 and use a product construction to create a new set of states that will accept the intersection (the PDA transitions will use a stack, and the DFA transitions won't because they don't need to). Ultimately, only one stack is used, so this won't be a problem. We thus simulate the DFA and PDA in parallel using the stack as required for the PDA. Our accept states are products of accepting states for the input PDA and DFA, with the added restruction that we must be at the bottom of the stack.