CS 181 - Homework 7

Kuruvilla Cherian, Jahan
UID: 104436427
Section 1B

November 17, 2017

Problem 2.11

To convert a CFG to a PDA, we create a stack as by pushing on the end symbol $\perp$ and the start non-terminal E. We then repeat the following steps continuously, while looking at the input:

1. If the top of the stack is E, pop it off and non-deterministically push one of its productions: $E + T$ or T.
2. If the top of the stack is T, pop it off and non-deterministically push one of its productions: $T \times F$ or F.
3. If the top of the stack is T, pop it off and non-deterministically push one of its productions: *(E)* or a.
4. If the top of the stack is a terminal symbol, compare it to the next input symbol. If they match repeat, else reject the current branch of non-determinism.
5. If the top of the stack is $\perp$, enter the accept state, which accepts if the entire input has been read.

Problem 2.20

We start by constructing two PDAs. One for A, called P that recognizes A, and one for A/B called *P'*. *P'* mimics P by reading symbols and branching non-deterministically just like P does. The difference comes when *P'* thinks it has reached the end, it stops reading more symbols, and instead begins simulating additional symbols as if they were in P by using a *DFA* for B. If *P'* simultaneously reaches an accept state in both simulations then it enters an accept state.

Problem 2.30

a

Assume L is context free. Let p be the pumping length. Let $s = 0^p 1^p 0^p 1^p \equiv uvxyz$. We consider the possibility where v or y contain more than one type of alphabet symbol, in which case uv^2xy^2z would not contain symbols in the needed order, thereby not falling in the CFL L. If we consider the possibility where v and y contain one type of alphabet symbol, in which case uv^2xy^2z would contain an incorrect number o f 0's and 1's which does not fall in the CFL L. Because we can't pump s without violating the conditions, L is not context-free.

d

Assume L is context free. Let p be the pumping length. Let $s = a^p b^p \# a^p b^p \equiv uvxyz$. We can assume vxy lies on either side of $\#$. Then if we pump up uv^pxy^2z, then $t_i \neq t_j$ and is thus not in L. If we assume $\#$ is in vxy, then we have two possibilities. $\#$ lies in either v or y in which case, pumping down to uv^0xy^0z, would remove $\#$ symbol, which does not fall in L. If $\#$ lied in x or u then by pumping down as before, we would have an imbalance in the numbers of a's or b's, which would make $t_i \neq t_j$ which is also not in L. Because we can't pump s without violating the conditions, L is not context-free.

Problem 2.45

Assume L is context free. Let p be the pumping length. Let $s = 1^{2p}0^p1^p1^{2p} \equiv uvxyz$ and in L, satisfying the three conditions of the pumping lemma. If we have vxy in the last two-thirds of this string, then pumping up to uv^2xy^2z would make it such that the first third has 0's but the last third doesn't, which doesn't belong in L. If however, vxy lies intersecting the first third, then it cannot extend beyond the first half of s. In this instance, v can contain both 0 and 1 in which case pumping up would make the string contain 0's in the first third but not the last third, thus not in L. This case is the same if v was empty and y contained both 0 and 1. The final case is if v contained only 1's or was empty and y contained only 1's in which case pumping up would make the string contain 0's in the last third but not the first third which makes it not in L. Thereby, L is not a context free language.