CS 361 – Homework 6

Total possible points: 60

1. (15 points) Use a **set notation** to **define** the language generated by the following grammar G

   S→TT|U

   T→0T|T0|#

   U→0U00|#

   $L(G) = \{w \text{ over } \Sigma |$if w contains one #, the right side of # will contain twice as many 0's as the left side of the #; if w contains two #'s, w can contain any number of zeros; w can only have one or two #'s}

   Consider the following strings. If they are in L(G), derive them.

   a) 00#0000

   $$R \Rightarrow U \Rightarrow 0U00 \Rightarrow 00U0000 \Rightarrow 00\#0000$$

   b) ϵ

   Not in L(G)

2. (15 points) Let A = $\{0^n1^m0^n1^m \mid m, n > 0 \text{ and } m \neq n\}$. If A is a context free language, then build the corresponding PDA or CFG. If not, prove that B is not a context-free language using the pumping lemma.
   a. For the purpose of contradiction we assume that A is context-free.
   b. Let's pick a string **s = $0^p1^{p+1}0^p1^{p+1}$**, $s \in A$ and $|s| = p + p+1 + p + p+1 \geq p$
   c. Let's consider all possible decompositions of s into wvxyz, in particular vxy substring, where $|vy| \leq p$. We have total of seven decomposition cases:
      i. **Case 1:** vxy is in the first 0^p, let v and y have the total of k 0's, where $p \geq k \geq 1$, then choosing i=2, we have a new string $wv^2xy^2z = 0^{p+k}1^{p+1}0^p1^{p+1}$, which is not in A since the number of the first 0's is always greater than the number of the second 0's, i.e., p+k > p
      ii. **Case 2:** vxy is in between the first 0^p1^{p+1}, let v and y contain together k 0's and m 1's, where k, m > 0 and $p \geq k+m$. Then choosing i=0, we have a new string $wv^0xy^0z = 0^{p-k}1^{p+1-m}0^p1^{p+1}$, which is not in A since the number of the first 0's is always less than the number of the second 0's, i.e., p-k < p
      iii. **Case 3:** :vxy is in the first 1^{p+1}, let v and y have the total of m 1's, where $p \geq m \geq 1$, then choosing i=2, we have a new string $wv^2xy^2z = 0^p1^{p+1+m}0^p1^{p+1}$, which is not in A since the number of the first 1's is always greater than the number of the second 1's, i.e., p+m > p
      iv. **Case 4:** vxy is in between $1^{p+1}0^p$, let v and y contain together m 1's and k 0's, where k, m > 0 and $p \geq k+m$. Then choosing i=0, we have a new string $wv^0xy^0z = 0^p1^{p+1-m}0^{p-k}1^{p+1}$, which is not in A since the number of the first 0's is always greater than the number of the second 0's, i.e., p > p – k

v. **Case 5:** vxy is in the second 0^p, let v and y have the total of k 0's, where $p \geq k \geq 1$, then choosing i=2, we have a new string $wv^2xy^2z = 0^p1^{p+1}0^{p+k}1^{p+1}$, which is not in A since the number of the first 0's is always less than the number of the second 0's, i.e., $p < p + k$

vi. **Case 6:** vxy is in between the second 0^p1^{p+1}, let v and y contain together k 0's and m 1's, where k, m > 0 and $p \geq k+m$. Then choosing i=0, we have a new string $wv^0xy^0z = 0^p1^{p+1}0^{p-k}1^{p+1-m}$, which is not in A since the number of the second 0's is always less than the number of the first 0's, i.e., $p-k < p$

vii. **Case 7:** vxy is in the second 1^{p+1}, let v and y have the total of m 1's, where $p \geq m \geq 1$, then choosing i=2, we have a new string $wv^2xy^2z = 0^p1^{p+1}0^p1^{p+1+m}$, which is not in A since the number of the first 1's is always less than the number of the second 1's, i.e., $p < p + m$

d. We have showed that for all possible decomposition we can always find i for which the new string is not in the language.

e. Therefore our assumption is wrong and A is not context-free.

Another possible solution

Let $A = \{ 0^n 1^m 0^n 1^m \mid m, n > 0 \text{ and } m \neq n \}$. Assume that A is context-free to obtain a contradiction. Let *p* be the critical length of the pumping lemma.

Consider the string: $w = 0^p 1^{p+1} 0^p 1^{p+1}$

We can write: $w = uvxyz$, with lengths $|vxy| \leq p$ and $|vy| \geq 1$

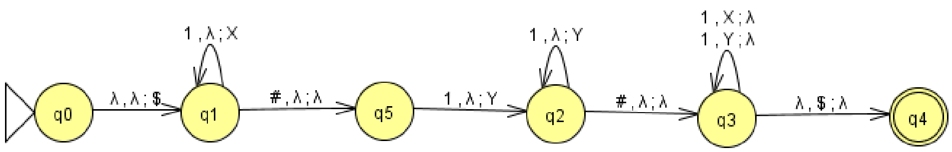
We now examine all the possible locations of string *vxy* in *w*.

1) vxy is in the first 0^p $v = 0^{k_1}$ $y = 0^{k_2}$ $k_1 + k_2 \geq 1$ When we pump p to 2, this generates: $uv^2xy^2z = 0^{p+k_1+k_2} 1^{p+1} 0^p 1^{p+1} \notin A$ We know that k1 or k2 has to be at least 1, so the length of the sequence of 0's will not be equal to the length of the sequence of 0's in the second 0^p. This is a contradiction. In the other locations where vxy only contains one alphabet symbol (1^{p+1}, 0^p, 1^{p+1}), we also obtain a contradiction.
2) vxy contains different alphabet symbols (across the first $0^p 1^{p+1}$) $v = 0^{k_1}$ $y = 1^{k_2}$ $k_1 + k_2 \geq 1$, where $|vxy| \leq p$ and $|vy| \geq 1$ When we pump p to 2, this generates $0^{p+k_1} 1^{p+k_2} 0^p 1^{p+1} \notin A$ We know that either k1 or k2 has be at least 1, so the first sequence of 0's won't match the second sequence of 0's, or the first sequence of 1's won't match the second sequence of 1's, or both. This is a contradiction. In the other valid locations where vxy contains different alphabet symbols ($1^{p+1} 0^p$, $0^p 1^{p+1}$), we also obtain a contradiction.
3) vxy overlaps different alphabet symbols (across the first $0^p 1^{p+1}$) $v = 01$ $y = 1^k$ provided that $|vxy| \leq p$ and $|vy| \geq 1$ When we pump p to 2, this generates $0^{p-1}(0101) 1^{p+k} 0^p 1^{p+1} \notin A$ The string we generate breaks the order of the alphabet A, and thus is a contradiction. In the other valid locations where v and/or y overlap different alphabet symbols ($1^{p+1} 0^p$, $0^p 1^{p+1}$), we also obtain a contradiction.

In all cases, we obtained a contradiction. We can conclude that *A* is not context-free.

3. (15 points) Let B = {$a\#b\#c$ | a, b, c are sequences of 1's; $|c|=|a|+|b|$; $|a|\geq 0$; and $|b|>0$ }. If B is a context free language, then build the corresponding CFG. If not, use the pumping lemma to show that B is not a context-free language.

B is context-free language. Below are its PDA & CFG:



S->X
X-> aXc | #Y
Y-> bYc|Z
Z->b#c

4. (15 points) Let C = {$a^n b^m c^{2n}$| m, n > 0}. If C is a context free language, then build the corresponding PDA. If not, use the pumping lemma to show that C is not a context-free language.

C is context-free language. Below is its PDA: