CS21004 - Tutorial 7

11th Mar 2019

Instructions : For the problems with (To submit), please write the answers neatly in loose sheets with your name and roll number. Submit to the TA at the end of the class.

1. Show that following languages are not context-free using pumping lemma

   (a) $L_1 = \{a^n b^j c^k : k > n, k > j\}$
   Solution: Let n be the pumping-lemma constant and consider string $z = a^n b^n c^{n+1}$. We may write z = uvwxy, where v and x, may be pumped up or down and $\mid vwx \mid \leq n$.
   If vwx does not have c's, then it contains only a and b. Then v and x has to be either a or b, If either of them are pumped up, then number of c no longer remains greater than that of a and that of b, and it could not be in the language. If vwx has a c, then it could not have an a, because its length is limited to n. Thus, uwy has n a's. For vwx, if c is trivial, then we can not pump up as number of b exceeds that of c. If c is non trivial, then we can not pump down as then number of c wil be less than or equals to the number of a and could not be in the language. Thus, we have a contradiction no matter how z is broken into uvwxy.

   (b) $L_2 = \{a^n b^j : n \leq j^2\}$ (To submit)
   Solution: Let m be the pumping-lemma constant and consider string $z = a^{m^2} b^m$. We have that $z = a^{m^2} b^m =$ uvxyz, with $\mid vxy \mid \leq m$ and $\mid vy \mid > 1$. Examine all the possible cases for the position of string vxy. First note that the string v cannot span simultaneously both a^{m^2} and b^m , since if we pump up v (repeat v), the resulting string is not in the language (a's are mixed with b's). Therefore, it must be that v is either within a^{m^2} or within b^m . The same holds for y. Below are the rest of the cases. Notice that in all cases we obtain a contradiction, and therefore the language L is not context-free.

      i. v is within a^{m^2} and y is within b^m
      We have that $v = a^k$ and $y = b^l$, with $1 \leq k + l \leq m$ since $\mid vxy \mid \leq m$ and $\mid vy \mid \geq 1$).
      Consider the case wherel $l \geq 1$. From the pumping lemma we have that $uv^0xy^0z \in L$. Therefore, $a^{m^2-k}b^{m-l} \in L$, and thus it must be that $m^2 - k \leq (m - l)^2$. However, this is impossible since:
      $(m - l)^2 \leq (m - 1)^2$ [since $l \geq 1$] $= m^2 - 2m + 1 < m^2 - k$ [since $k \leq m$]
      Consider the case wherel $l = 0$. It must be that $k \geq 1$ [since $k + l \geq 1$]. From the pumping lemma we have that $uv^2xy^2z \in L$. Then $a^{m^2+k}b^m \in L$, which is impossible since $m^2 + k > m^2$

      ii. v and y are within a^{m^2}
      If we pump up v and y (repeat them) , we obtain a string of the form $a^{m^{2+k}} b^m$, with $k \geq 1$, which obviously is not in the language.

      iii. v and y are within b^m If we pump down v and y (remove them) , we obtain a string of the form $a^{m^2} b^{m-k}$, with $k \geq 1$, which obviously is not in the language.

   (c) $L_3 = \{a^n b^j c^k : k = jn\}$ (Home)

2. Covert the following grammars into Chomsky Normal Form.

   (a) $S \to aXbX$
   $X \to aY|bY|\epsilon$
   $Y \to X|c$ (To submit)
   Solution : Corresponding to P, $\hat{P}$ is,

   $S \to aXbX|abX|aXb|ab$
   $X \to aY|bY|a|b|\epsilon$
   $Y \to X|c$
   $\Downarrow$

$S \to aXbX|abX|aXb|ab$
$X \to aY|bY|a|b$
$Y \to X|c$
$\Downarrow$
$S \to aXbX|abX|aXb|ab$
$X \to aY|bY|a|b$
$Y \to aY|bY|a|b|c$
$\Downarrow$
$S \to EF|AF|EB|AB$
$X \to AY|BY|a|b$
$Y \to AX|BY|a|b|c$
$E \to AX$
$F \to BX$
$A \to a$
$B \to b$
$C \to c$

(b) $S \to AACD$
$A \to aAb|\epsilon$
$C \to aC|a$
$D \to aDa|bDb|\epsilon$
(Home)
Solution :In this case, $\hat{P}$ becomes:

$S \to AACD|ACD|AAC|CD|AC|CaC|a$
$A \to aAb|ab|\epsilon$
$C \to aC|a$
$D \to aDa|bDb|aa|bb|\epsilon$

Step 1: **Eliminate ϵ-production**: The nullable variables are A and D. So remove those production.

$S \to AACD|ACD|AAC|CD|AC|C$
$A \to aAb|ab$
$C \to aC|a$
$D \to aDa|bDb|aa|bb$

Step 2: **Eliminate unit-production**: We remove the unit production $S \to C$ replacing it by $S \to aC|a$.

$S \to AACD|ACD|AAC|CD|AC|aC|a$
$A \to aAb|ab$
$C \to aC|a$
$D \to aDa|bDb|aa|bb$

Step 3: **Restricting the right side of production to single terminals or strings of two or more variables**:

$S \to AACD|ACD|AAC|CD|AC|X_aC|a$
$A \to X_aAX_b|X_aX_b$
$C \to X_aC|a$
$D \to X_aDX_a|X_bDX_b|X_aX_a|X_bX_b$
$X_a \to a$
$X_b \to b$

Step 4: **Final Step to CNF**: There are six production whose right sides are too long.

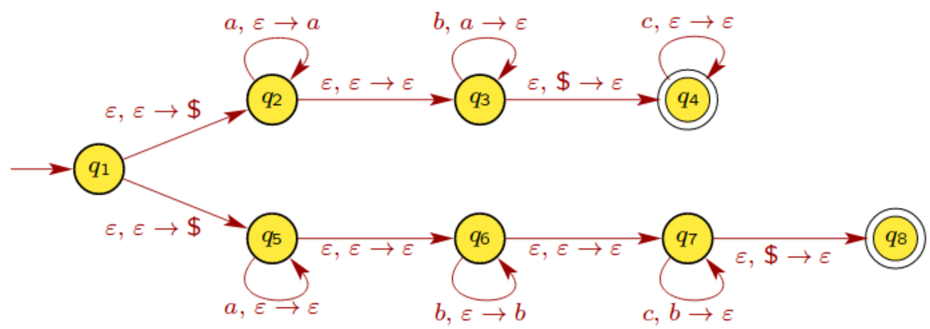
$S \to AT_1|AU_1|AV_1|CD|AC|X_aC|a$
$T_1 \to AT_2$
$T_2 \to CD$
$U_1 \to CD$

$V_1 \to AC$
$A \to X_aW_1|X_aX_b$
$W_1 \to AX_b$
$C \to X_aC|a$
$D \to X_aY_1|X_bZ_1|X_aX_a|X_bX_b$
$Y_1 \to DX_a$
$Z_1 \to DX_b$
$X_a \to a$
$X_b \to b$

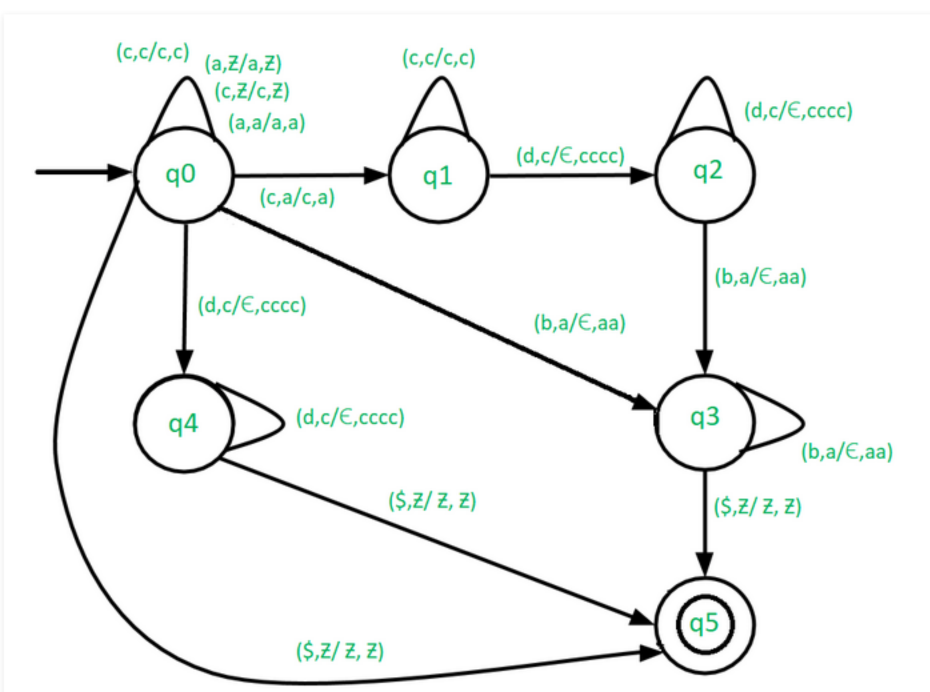
3. Construct the PDA for the following for the language over a, b)

   (a) L1 = $a^ib^jc^k|i,j,k \geq 0$, and i=j or j=k

   Solution :



   (b) L2 = $a^{2m}c^{4n}d^nb^m|m,n \geq 0$
   Solution :



   (c) L3= $x\$y|\exists n : x = binary(n) \wedge y = binary(n+1)$ where binary(n) is the binary encoding of natural number n. For example, this set contains $0\$1, 1101\$1100$ and $001\$101$ but not $1\$1 or 11\$10$. (Home)

Hint : The machine will read the binary encoding of n from the input, from least-significant digit to most-significant digit, at the same time pushing the binary encoding of n+1 onto its stack. When it encounters the $, it will match the stack against the remaining input.

4. Give CFG for the following languages.

   (a) Write a rudimentary CFG to parse the roman numerals $1-99(i, ii, iii, iv, v, ..., ix, x, ..., xl, ..., lxxx, ..., xc, ..., xcix)$. Consider the terminals c, l, x, v, i where $c = 100, l = 50, x = 10, v = 5, i = 1$. (Home)

   Solution :
   $S \rightarrow xTU|lX|X$
   $T \rightarrow c|l$
   $X \rightarrow xX|U$
   $U \rightarrow iY|vI|I$
   $Y \rightarrow x|v$
   $I \rightarrow iI|\epsilon$

   (b) Construct a context free grammar for generating regular expressions which has the set of terminals T $= a, b, ep, +, *, (,)$ over a, b, with + meaning the RegExp OR operator, and ep meaning the the ϵ symbol. (Home)