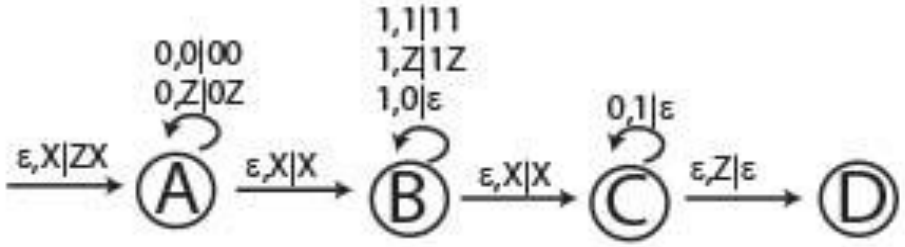
Name_____

CS 383
Exam 2 Solutions
November 2019

1. Which of the following languages are context-free? Read the descriptions carefully. Write "C" next to the languages that are context-free, "N" next to the ones that are not. No proofs are necessary.
   a. __**C**___ $\{0^n1^n1^n \mid n >= 0\}$ This is 0^n1^{2n}. S=>0S11 | e will generate it.
   b. __N____ $\{0^n1^n1^n0^n \mid n>=0\}$ This is $0^n1^{2n}0^n$ A pumping lemma argument shows it isn'r CF
   c. __C___ $\{0^n1^m \mid$ n and m are either both even or both odd$\}$ This is Regular
   d. ___C___Strings of the digits 0-9 whose digits sum to an even number, such as 24473 or 112233. This is Regular. Have states that track whether the sum so far is even or odd.
   e. __C___ $\{0^n1^n0^m1^m \mid n>0$ and $m>0\}$ This is the concatenation of 2 CF languages.
   f. ___C__ $\{0^n1^n \mid n > 0$ and n is odd$\}$ This is the intersection of 0^n1^n (Context-Free) and $\{0^n1^m \mid$ n and m are odd$\}$ which is Regular.
   g. ___N__ Strings of the form vcv where v is a string of 0s, 1s, and 2s (and c is just the letter c), such as 0210c0210. This is just like {vv} which we showed in class is not C-F.
   h. __C__ Strings of the form vcw where v and w are both strings of 0s, 1s, and 2s (and c is just the letter c), where v and w have the same length. 1210c2020 is such a string. Make a PDA that pushes A on any digit before c, then pops A on any digit after c.
   i. ___C__ Strings of the form vcw, where v and w are both strings of 0s, 1s ,and 2s (and c is just the letter c), such that the digits of v sum to the same value as the digits of w. For example, 012011c221 is such a string because the digits before and after c both sum to 5. Before c push d As on digit d. After c pop d As on digit d.

2. Construct a PDA that accepts by final state the language $\{0^n1^{n+m}0^m \mid m>= 0, n >= 0\}$

   The following uses "Z" as the stack bottom symbol.



3. Here is a grammar:
   S => 0A2 | BC
   A => 0A2 | 02
   B =>0B | 0
   C => C2 | 2

   a. Use this grammar to construct either a parse tree or a derivation (your choice; one is about as easy or hard as the other) for the string 00022.

      S => BC
      => 0BC
      => 00BC
      => 000C
      => 000C2
      => 00022

   b. Find a string that has two completely different parse trees (or derivations) with this grammar,

      This is ambiguous for every string 0^n2^n, such as 0022:

      S => 0A2
      => 0022

      S => BC
      => 0BC
      => 00C
      => 00C2
      => 0022

4. Convert the following grammar to Chomsky Normal Form:

   A => 0A2| BC

   B => 0B2 | C | ε

   C => 1A1 | 1

   Step 1: B is nullable, so we modify all rules containing B:

   A => 0A2 | BC | C

   B => 0B2 | 02 | C

   C => 1A1 | 1

   Step 2: All symbols are reachable and generating.

   Step 3: Remove unit rules

   A => 0A2 | BC | 1A1 | 1

   B => 0B2 | 02 | 1A1 | 1

   C => 1A1 | 1

   Step 4: Break down to binary rules

   A => ZA_1 | BC | NA_2 | 1

   A_1 => AT

   A_2 => AN

   Z => 0

   N => 1

   T => 2

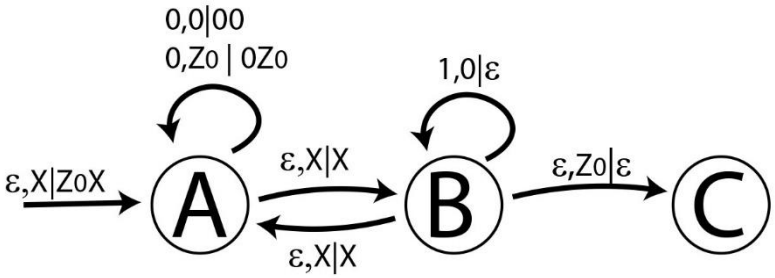
   B => ZB_1 | ZT | NA_2 | 1

   C => NA_2 | 1

5. Give a careful pumping lemma proof that $\{1^n2^n3^m \mid n>0,\ m>n\}$ is not a context free language. If you aren't clear about the language, it is the subset of 1*2*3* with the same number of 1s and 2s, and more 3s.

   Suppose this language is context-free. Let p be its pumping constant. Let z be the string $1^p2^p3^{p+1}$, which is certainly longer than p. Consider any decomposition z=uvwxy, where |vwx| <= p and v and x aren't both empty. Since |vwx| <= p, v and x can together contain at most 2 of the 3 digits. If v and x contain no 3s, then they must contain either 1s or 2s (or both), so uv^2wx^2y does not have more 3s than 1 and 2s. If v and x do contain at least one 3, then v and x contain no 1s, so uv^0wx^0y contains the same number of 1s as z and fewer 3s than z, which means that uv^0wx^0y does not have more 3s than 1s. Either way we have found a value of n for which uv^nwx^ny is not in the language. This contradicts the pumping lemma, so the language must not be context-free.

6. In class we developed an algorithm by Noam Chomsky that constructs a grammar equivalent to a given PDA. Apply this algorithm to the following PDA and give the derivation in this grammar of the string 001011. Note that the PDA accepts by empty stack.



S => [AZ_0C]
=> 0[A0B] [BZ_0C]
=> 00[A0B][B0B] [BZ_0C]
=> 00[B0B][B0B] [BZ_0C]
=> 001[B0B] [BZ_0C]
=> 001[A0B] [BZ_0C]
=> 0010[A0B][B0B] [BZ_0C]
=> 0010[B0B][B0B] [BZ_0C]
=> 00101[B0B] [BZ_0C]
=> 001011[BZ_0C]
=> 001011