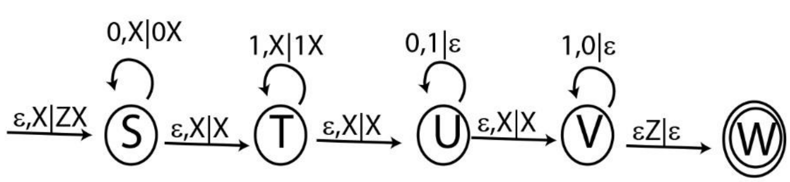
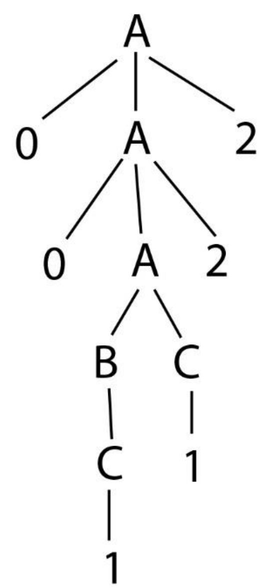
CS 383
Exam 2 Solutions
November 17, 2017

1. Which of the following are context-free? Write "CF" next to the languages that are context-free, "N" next to the ones that are not. No proofs are necessary.
   a. $\{0^n1^n0^n \mid n >= 0\}$ N
   b. $\{0^n1^n1^n \mid n >= 0\}$ CF; this is $\{0^n1^{2n}\}$
   c. $\{0^n1^n1^n0^n \mid n >= 0\}$ N this is $\{0^n1^{2n}0^n\}$
   d. Palindromes of 0s and 1s CF
   e. The language of balanced parentheses: The alphabet for this language has just two characters: a left parenthesis and a right parenthesis. Strings in the language have the same number of left and right parentheses, and every prefix of such a string has at least as many left parentheses as right ones. (() ()) is in this language but ( ))(() is not. CE
   f. {vw | v is a string of 0s and 1s with length 3 or more and w is the first 3 letters of v} CF; it is even Regular
   g. $\{w0^n \mid$ w is a string of 0s and 1s and n is the length of w$\}$ CF
   h. {uvw | u,v,w are all strings of 0s and 1s with the same length } CF; thus is aiso regular. It is the set of strings whose lengths are divisible by 3.

2. Construct a PDA that accepts by final state the language $\{0^n1^m0^m1^n \mid n >= 0 \}$



3. Here is a grammar, Construct a parse tree for this grammar for the string 001122:
   A ➔ 0A2| BC
   B ➔ 0B2 | C
   C ➔ 1A1 | 1



4. Convert the following grammar to Chomsky Normal Form:

   A ➔ 0A2| BC

   B ➔ 0B2 | C

   C ➔ 1A1 | 1

   Eliminate unit rules:

   > A ➔ 0A2 | BC
   >
   > B ➔ 0B2 | 1A1 | 1
   >
   > C ➔ 1A1 | 1

   ****

   Eliminate terminal symbols

   > A ➔ ZAT | BC
   >
   > B ➔ ZBT | NAN | 1
   >
   > C ➔ NAN | 1
   >
   > Z ➔ 0
   >
   > N ➔ 1
   >
   > T ➔ 2

   ****

   Factor into 2 symbols per rule

   > A ➔ ZA_1 | BC
   >
   > A_1 ➔ AT
   >
   > B ➔ ZB_1 | NN_1 | 1
   >
   > B_1 ➔ BT
   >
   > N_1 ➔ AN
   >
   > C ➔ NN_1 | 1
   >
   > Z ➔ 0
   >
   > N ➔ 1
   >
   > T ➔ 2

5. Give a careful pumping lemma proof that $\{0^i1^j2^k \mid i > 1, 0<j,k<i\}$ is not a context free language. If you aren't clear about the language, it is the subset of 0*1*2* where there are more 0s than either 1s or 2s.

   Suppose the language is context-free. Let p be its pumping constant. Consider the string $z=0^{p+1}1^p2^p$. Now consider any decomposition z=uvwxy where vx isn't empty and $|vwx| < p$. If we are going to pump v and x they each need to consist of only one mind of symbol, so at least one of the three digits will not be in vx. If 0 is the omitted digit then uv^2wx^2y increases the number of 1s or 2s but not the number of 0s, so it will no longer have more 0s than 1s or 2s. On the other hand, if 0s are not omitted from vx then 1s or 2s must be, and uv^0wx^0y decreases the number of 0s while either not changing the number of 1s or not changing the number of 2s. So uv^0wx^0y is not in the language. No matter how z is decomposed, there are values of n for which uv^nwx^ny is not in the language. We found a long string in the language that is not pumpable, so the language can't be context-free.

6. Both parts of this question concern the language $\{0^d1^n \mid d \text{ is a divisor of } n\}$. As one example 0^31^{15} is a string in this language because 3 divides evenly into 15. Part (a) asks for an English description to help me understand your part (b).

   a. Describe in English the steps a Turing Machine goes through to accept this language0
      i. For each 0 in the string: overwrite the 0 with X, then overwrite a 1 with Y. When there are no more 0s look past all of the Ys. If there is a 1 go to step (ii); if there is a B (blank) accept
      ii. Go back and overwrite all Xs with 0s. Go back to step (i)

   b. Construct a Turing Machine (with states and transitions) that implements your algorithm from (a).