CS 2200 Spring 2019 HW 05

Dr. George Markowsky
Computer Science Department
Missouri Science & Tech

Due Tuesday,April 2, 2019, 11:59PM

Submit your HW as a SINGLE PDF file that includes your answers, your programs, and their output. Also include any data files if you used them. Please make sure that your PDF is well-organized so we can quickly find all the pieces of each problem that belong together.

PROBLEMS

1. (10 points) Find a deterministic finite-state automaton that recognizes the language, L, consisting of all strings in {a, b}* that contain an odd number of b's such that there is at least one a between every two b's in the string. Verify your solution by running some non-trivial cases on your simulator. Include your graphviz drawing, your .gv file, and your .ndfsa file.

2. (5 points) Construct a regular expression that generates the language, L, defined in Problem 1.

3. (10 points) Let A = {c,d}. Let $L_2 \subseteq A^*$ be the language consisting of all strings not in the language, L_3, which is generated by the regular expression c*d*c*. Find a deterministic finite-state automaton that recognizes L_2. Verify your solution by running some non-trivial cases on your simulator. Include your graphviz drawing, your .gv file, and your .ndfsa file.

4. (5 points) Find a regular expression that generates the language, L_2, defined in Problem 3

5. (10 points) Let $L_3 \subseteq \{f, g\}^*$ be the language of all strings generated by the regular expression (fg + ffg + fgf)*. Construct a non-deterministic finite-state machine that recognizes L_3. Verify your solution by running some non-trivial cases on your simulator. Include your graphviz drawing, your .gv file, and your .ndfsa file.

6. (10 points) Using the subset construction described in class, construct a deterministic finite-state machine based on the non-deterministic finite-state machine that

you constructed in Problem 5. Verify your solution by running some non-trivial cases on your simulator. Include your graphviz drawing, your .gv file, and your .fsa file.

7. (10 points) Let G be the context-free grammar { {Q,V}, {c,d},{ $Q \to QQ|V$, $V \to cVd|cd$ }, Q}.
   (a) (5 points) Describe the language L_6 = L(G). Prove that G is ambiguous.
   (b) (5 points) Give an unambiguous grammar H, such that L(H) = L(G). Give some justification for the claim that H is unambiguous.

8. (20 points) Consider the CFG G = { {S}, {a, b}, {$S \to aSbS|bSaS|\epsilon$ }, S }. Prove by induction that L(G) = { w $\in$ { a, b}* | the number of a's in w = the number of b's in w }. Note that there are two things to prove here. First, you have to prove that if a string of terminals is generated by G, then the number of a's in it is equal to the number of b's. Second, you have to prove that if a string of a's and b's has an equal number of a's and b's, then it can be generated by G.

9. (20 points) Let G = (Vars, Alph, Rules, Start) be a CFG as described below. Using the algorithms discussed in class, convert G into an equivalent grammar that is in Chomsky Normal Form.
   (a) Vars = { P, Q, R, S, T, U. V. W. X, Y, Z }.
   (b) Alph = { a, b, c }.
   (c) Start = S.
   (d) Rules =
      i. $S \to aXT|YbT|UbZ|UWc|PQT$.
      ii. $P \to aT$.
      iii. $Q \to QT|aQ$.
      iv. $T \to cT|\epsilon$.
      v. $U \to aU|\epsilon$.
      vi. $X \to aX|R|\epsilon$.
      vii. $R \to aRb|\epsilon$.
      viii. $Y \to Yb|R|\epsilon$.
      ix. $V \to bVc|\epsilon$.
      x. $W \to Wc|V|\epsilon$.
      xi. $Z \to bZ|V|\epsilon$.