Jay Patel
CS 331
HW 10 Extra Credit
Professor Oliver

Q1.

Steps:

“On input <M> where M is a PDA:

1. Firstly, convert into CFG from M
2. Using PL, let p be the PL of the CFG, PDA p accepts many strings infinitely if it accepts any string > p.
3. All the strings > p, let R be the regular language which accepts it
4. Let C be the intersection of a CFG and a regular language in a CFG.
5. If L(C) empty, accept else reject

Q2.

Lets assume that EQ_{DFA} = {<A, B>|A, B are DFAs and L (A) = L (B)} and suppose M is a turning machine for determining the decidability of EQ_{DFA}

Steps:

M = “on input <A, B> where A and B are DFAs

1. Calculate number of states
2. Iterate to all the string which is under Σ
3. For each string w
    a. Simulate DFA A on string w
    b. Simulate DFA B on string w
        i. We know that machine M is running as a decider right now
4. After all the n, m strings accept it else reject
5. If M accepts, accept it
6. If M reject, reject it

Size for working:
DFA does not accept the same language that’s why we checked for first n, m strings. So the string w of size |w| and it is less than equal to size of n,m for A (w) ≠ B (w)
Using contradiction, suppose that the first string provides a different output of DFAs A and B is w’ the length of l of |w’| > n, m

Q3.

Assume a TM K that is used to decide E_{CFG}

Steps:

1. A PDA P is constructed in such a way that L (P) = {w|w is a palindrome}
2. A PDA P' is constructed in such a way that L (P') = L (P) ∩ L (M)
3. Now P' is converted into an equivalent CFG G
4. Here, TM K is used to check if L (G) is empty
5. If L (G) is empty the reject, else accept

Q4.
Consider the following proof, which shows that C is decidable

Steps:

1. A DFA A is constructed in such a way that it used to recognize that language of the regular expression
2. A DFA F is constructed which is used for the CFL L (G) ∩ L (A)
3. After performing simulation on the TM E_{CFG} $on\ L\ (F)$
4. If the above accepts, reject else accept

If it is an intersection of a regular language and a CFL we know that the language is CFL. Hence F will be CFG. So if G produces some string y with x as its substring the intersection L (F) should be non-empty.

Q5.
Consider a TM D as follows:

D = “m as input:
Steps:
1. Check m ∉ {0,1}* then reject
2. Check m is equal to any s_i
3. Use Enumerator E to enumerate <M1> <M2> <M3>…. <Mn>
4. Execute M1 having input m
5. Check whether M1 is accepts. If it accepts then reject. Otherwise accept
6. As it is already mentioned D is a TM, which act, as decider.
7. D is different from M1, so A does not contains <D>