CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Name: \_\_\_\_Derek Trom\_\_\_\_\_\_\_

**Home Assignment 6: 90 points + 20 points (optional)**

In any (N/D)PDA, assume that a start stack symbol z is already in the stack; so, you don’t have to insert z into the stack at the beginning of transition.

Q1.[30] Prove if the following languages are CFL or not.

If L is a CFL, give its CFG. Otherwise, prove it by Pumping Lemma.

If any closure property of CFL is applicable, apply them to simplify it before its proof.

1. [10] L = {*wwRw* | *w* ∈ {*a, b*}\*}
   1. Not a CFL
   2. Assume that the language is context free
   3. Z = abbaab
   4. For w = uvxyz lets use u=a,v=b,x=ba,y=a,z=b
   5. If for any value of i, uvixyiz belongs to L then L is Context Free
   6. Let’s use i=2, the abbbaaab which is not part of the language
   7. Thus this contradicts our statement and is not context free
2. [10] L = { *anwwRbn* | *n* ≥ 0, *w* ∈ {*a, b*}\*}
   1. S→aSb|aAa|bAb|λ
   2. A→aAa|bAb|λ
3. [10] L = {*anbjajbn* | *n* ≥ 0, *j* ≥ 0}
   1. This is a CFL, its CFG is:
      1. S→ aSa | aAa | A | λ
      2. A→ bAa | λ
4. [10, optional] L = { *an*| *n* is a prime number }
   1. Let m be the parameter of the pumping lemma. Let p be a prime such that p >= m.
   2. We choose to pump the string ap ∈L.
   3. Since ap = uvxyz, we have that v = ak and y = al , with k+l >= 1 (since |vy| >= 1).
   4. From the pumping lemma we have that uv1+pxy1+pz ∈ L, and therefore ap+kp+lp ∈ L.
   5. Subsequently, ap(1+k+l) ∈ L, which is impossible since p(1+k+l) is not a prime.
   6. Thus, we have a contradiction and the language L is not context-free.

Q2. [20] Prove that the following languages are linear or not.

If L is linear, give the linear-CFG for L. Otherwise, prove it by Pumping Lemma for a Linear-CFL.

1. [10] L = { *w* | *na*(*w*) + *nb*(*w*) = *nc*(*w*) } is not linear.
   1. Assume L is LCFL
   2. z = cmambmcm
   3. u = c, v = a, x = λ, y = b, z = c
   4. i = 0 with uvixyiz then, cc which is not part of the language
   5. Thus L is not a linear CFL as pumping lemma fails
2. [10] L = { *anbmcn* | *n, m* ≥ 0 } ∪ { *anbncm* | *n, m* ≥ 0 } is linear or not.
   1. The first language of the union is in fact a linear language
   2. S → aSc | bA | λ
   3. A → bA | λ
   4. However the second is not linear
   5. As shown by pumping lemma:
   6. Given the magic number m, the string ambmcm
   7. If we choose i = 0 we get the am-1bmcm-1
   8. The family of linear languages have closure under union with each other
   9. But not with other languages thus is it cannot be regular.

Q3. [30] Prove the following properties clearly.

1. [10] The family of CFLs is closed under reversal.
   1. Let G be the context-free grammar that generates the language L. We construct from G a new grammar G’ as follows. For every production X → v of G, we add the production X →vR in G’ , where X is a variable, and v is a string of terminals and variables. It is easy to see that that a string w is generated by grammar G if and only if the string wR is generated by grammar G’ . Therefore, grammar G’ generates the language LR , and thus the language LR is context free
2. [10] The family of DCFL is closed under regular difference:

i.e. for a DCFL L1 and a RL L2, L1 − L2 ∈ DCFL.

* + - 1. L1=DCFL, L2=REGULAR, L1-L2= L1 ∩ L2' =L1-(L1 ∩ L2). Now L2 is closed under Compliment which means L2' is regular. Now L1 is DCFL. As I said above if language is regular it is also DCFL. Now if we use Intersection will be there in regular and dcfl it is Dcfl(The intersection part or the common part). An example of this. Assuming L1={anbm for all n=m} L2={an for all n>=1}. So L1-L2 is DCFL.

1. [10] The family of CFLs is not closed under complement. Give an example for it.
   1. Assume the complement of every CFL is a CFL. Let L1 and L2 be 2 CFLs. Since CFLs are close under union, and we are assuming they are closed under complement, = L1 ∩ L2 is a CFL. However, we know there are CFLs whose intersection is not a CFL. Therefore, our assumption that CFLs are closed under complement is false.
2. [10] If L1 is linear and L2 is regular, L1⋅L2 is a linear language.
   1. All regular languages have a linear grammar that describes the language. So let’s assume there is a linear grammar that represents L1 and has a start symbol S. Because every regular language is also a context free language we can also assume that a grammar representing L2 would be context free. We can generate another grammar to represent L2. This grammar however can be described as a left linear language. If we combine the two languages by putting S in front of any productions with only a nonterminal like A → Sa from A → a this will produce L1⋅L2 as a linear grammar.
3. [10, optional] The family of DCFLs is **not** closed under reversal. Give an example.
   1. L = { *(a+b)\*w*bwR | *w* ∈ {*a, b*}\*} this is a nondeterministic CFL because the start of the string after (a+b)\*, *w*bwR , is unable to be decided. This makes it nondeterministic because PDAs read from left to right.