CSci 435: Formal Languages and Automata

Instructor: Dr. M. E. Kim Name: Sam Dressler

**Home Assignment 6: 69/90 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.[23/30] 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. [10/10] L = {*wwRw* | *w* ∈ {*a, b*}\*}

Assume L is a context free language.

Then there exists a pumping length P = 3

Let w = aPbP

So, string S = a3b3 b3a3 a3b3

S = aaabbb bbbaaa aaabbb

We can divide S into 5 parts u,v,x,y, and z

u = aaabbb, v = b, x = b, y = b, z =aaaaaabbb

uv1xy1z = aaabbbbbbaaaaaabbb

uv2xy2z = aaabbb bbbbbaaa aaabbb

numbers of a’s and b’s in non-pumped w = 3

since numbers of a’s and b’s in wR is 3 and 5 respectively, S is not in L

Therefore, L is not a context free language due to the language not passing rule 1.

1. [0/10] L = { *anwwRbn* | *n* ≥ 0, *w* ∈ {*a, b*}\*}

Assume L is a context free language

Then there exists a pumping length P = 3

Let w = aPbP and n = p

So, string S = a3a3b3b3a3b3 = aaaaaabbbbbbaaabbb

We can divide S into 5 parts uvxyz

u = aaa, v = aaa, x = bbbbbb, y = aaa, z = bbb

uv1 xy1z = aaaaaabbbbbbaaabbb

uv2xy2z = aaaaaaaaabbbbbbaaaaaabbb this satisfies rule 3 since the number of a’s and b’s are the same for w and wR

Let’s change how uvxyz are defined.

U = aa v =aaaa x = bbbbbb y = a, z = aabbb

Condition 1 is satisfied since |vy| >= 1 and Condition 2 is satisfied since |vxy| <= |s|

Then uv0xy0z = aabbbbbaabbb

This string does not satisfy rule one since it does not exist in the language L,

therefore, L is not a context free language.

The CFG for L is: S → *a*S*b* | S1,

S1 → *a*S1*a* | *b*S1*b* | λ .

So, L is a CFL.

1. [3/10] L = {*anbjajbn* | *n* ≥ 0, *j* ≥ 0}

A Context free language exists:

S-> aSb | bSa |λ S 🡪 bSa 🡪 b(aSb)a 🡪 baba ∉L

CFL: S → *a*S*b* | A; A → *b*A*a* | λ

1. [10/10, optional] L = {*an*| *n* is a prime number}

Assume L is a context free language

Then pumping length P = 7

So, string s = aP = aaaaaaa

Let u = a, v = aa, x = a, y = a, z = aa

Condition 1 is satisfied since |vy| >= 1 and Condition 2 is satisfied since |vxy| <= |s|

uv1xy1z = aaaaaaa

uv2xy2z = aaaaaaaaaa

since the number of a’s is 10, the string does not contain a prime number of a’s and s is not included in the language L. Therefore, L is not a context free language because it does not pass the third rule.

Q2. [8/20] 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. [8/10] L = {*w* | *na*(*w*) + *nb*(*w*) = *nc*(*w*)} is not linear.

L is not linear since the number of a’s plus the number of b’s is equal to the number of c’s

Assume L is linear and apply pumping lemma

Let string s = aaabbbcccccc

Split s into 5 parts uvxyz

u = aa, v = a, x = bb, y =b, z = cccccc

s = uv1xy1z = aaabbbcccccc fits in language L

s = xv2xy2z = aaaabbbbcccccc

This string does not fit in the language L since the number of a’s and b’s does not add up to equal the number of c’s.

Use a Pumping Lemma for Linear language.

Let w = *amc2mbm*∈ L.

Let’s choose *u* = *ai*, *v* = *aj,* *x* = *am-i-j c2mbm-k-l*, *y* = *ak*, *z* = *al*, in the 1st *am* or in the last *bm* ,

then, x = *am-i-j c2mbm-k-l*, where *i, j, k, l* > 0 and let’s choose *i, j, k, l* ≤ *m.*

Then, |uvyz| ≤ *m*, |vy| ≥ 1.

Then, *w0* = uxz = *am-j cm2mbm-l* ∉L since (*m*-*j*) + (*m*-*l*) < 2*m* where *j, l* > 0.

So, L is not a linear CFL.

1. [10] L = {*anbmcn* | *n, m* ≥ 0} ∪ { *anbncm* | *n, m* ≥ 0 } is linear or not.

Assume L is linear and apply pumping lemma

Let string S = aaabbbcccc

Split s into 5 parts uvxyz

u = a, v = aa, x = bbb, y = c, z = c

Condition one is satisfied since |vy| = 3 >= 1

Condition two is satisfied since |uvyz| <= m where m >0.

To satisfy condition three, take an i = 2 so S = uv2xy2z = aaaaabbbccc

S is hence not in L since the number of a’s are not equal to the number of b’s or the number of c’s.

This contradicts our claim that L is linear, therefore L is not linear.

L is a linear CFL whose linear CFG is:

S → A | B,

A → *a*Ac | C, C → bC | λ,

B → Bc | D, D → *a*Db | λ

Q3. [38/40] Prove the following properties clearly.

1. [8/10] The family of CFLs is closed under reversal.

We can show that a CFG is closed under reversal:

Consider the language L in CFL. This means that there is a grammar G that satisfies the CFL. For ever production V->AB in G, replace it with V1-> BA. We can form a parse tree for the derived string and see that the language derived will be the reverse of the initial languages since the new productions will be the reversal of the original.

Let L be a CFL generated by a CFG G.

Given aCFG G=(V, T, S, P), construct a CFG G’ = (V, T, S, PR) where

PR is defined from P by replacing every production A → x by A → xR where A ∈ V, x ∈ (V ⋃ T)\*

Then, show that L(G’) = LR.

🡨 ) For w ∈ L, w is derived by G: S ⇒\* x ⇒\* w where x∈ (V ⋃ T)\* .

For wR ∈ LR, S ⇒\* xR ⇒\* wR in G’ since the production of G’ , PR, is defined from P.

Thus, wR ∈LR = L(G’).

→ ) If w’ = wR ∈L(G’), i.e. w’ is generated by G’, S ⇒\* xR ⇒\* w’.

Then, (w’)R = w is derived by the production in G, s.t. S ⇒\* x ⇒\* w’R = wsince the production rule of G’ was defined in the reverse way of P.

So, w’ ∈ LR .

Thus, L(G’) = LR. Therefore, CFL is closed under reversal.

1. [10/10] The family of DCFL is closed under regular difference:

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

If we have a DCFL L1 and and a regular language L2. The difference L1 – L2 can be rewritten as the intersection of L1 and the complement of L2. Since L2 is a regular language, it is closed under theorem 4.1. Furthermore, we can see that L1 intersection L2 is closed by theorem 8.5 since L1 is a CFL and L2 is a regular language. Therfore the family of DCFL’s are closed under regular difference.

1. [5/10] The family of CFLs is not closed under complement. Give an example for it.

L = {anbmck | n, m, k ≥ 0 and n ≠ m or n ≠ k} is not closed under complement

L = {w1cw2 | w1, w2 ∈ {*a, b*}\*, w1 ≠ w2} is a CFL. But, LC is not a CFL.

Proof) Suppose LC  is a CFL.

Then, LC ∩ L((*a*+*b*)\*c(*a*+*b*)\*) = {wcw | w ∈ {*a, b*}\*}.

But, {wcw | w ∈ {*a, b*}\*} is not a CFL – Contradiction!

So, L = {w1cw2 | w1, w2 ∈ {*a, b*}\*, w1 ≠ w2} is a CFL, but LC  = {wcw | w ∈ {*a, b*}\*} ∪ {w| c ∉ w, w ∈ {*a, b*}\* } is NOT a CFL.

1. [8/10] If L1 is linear and L2 is regular, L1⋅L2 is a linear language.

Consider grammar G1 that produces L1. Since G1 is a linear grammar it will produce only languages that are also linear. If a regular language is concatenated to the end of the the language produced by L1 it will still be considered linear. This is because all regular languages formed by regular grammars are inherently linear, however the converse is not true. All regular languages are linear since any production of a regular grammar has the form A->u or A -> uXv but not A -> λ.

Let G1 = (V1, T, S1, P1) be a linear grammar for L1

And G2 = (V2, T, S2, P2) be a left-linear grammar for L2.

Construct a G2’ from G2 by replacing every production of the form V 🡪 *x*, *x* ∈ T\* with V 🡪 S1*x* -- P2’

So, the combined G = G1 ∪ G2’ = ( V1 ∪ V2 , T, S, P1 ∪ P2’) where S is a new start symbol and P2’ above.

Then, further show that S ⇒\* S1w ⇒\* *uw* if and only if u ∈ L1 and w ∈ L2.

1. [7/10, optional] The family of DCFLs is **not** closed under reversal. Give an example.

L = {WRcW (a+b+c)\* | W ∈ (a+b)+ } is not closed under reversal.

L = { *bnan*| *n* ≥ 0 } ∪ { *cb2nan* | *n* ≥ 0} is DCFL.

LR = { *anbn*| *n* ≥ 0 } ∪ { *anb2nc* | *n* ≥ 0} is Nondeterministic CFL.