|  |  |
| --- | --- |
| Probl  # | Pts |
| 1 | 40 |
| 2 | 25 |
| 3 | 25 |
| 4 | 33 |
| 5 | 41 |
| 6 | 20 |
|  |  |
| Sum | 100% |

GRADE CALCULATION = round (

(pts#1/.4+ pts#2/.25+ pts#3/.25+ pts#4/.33+ pts#5/.41+ pts#6/.2)/6, 0)

[40pts] Problem #1

Call R the set of ALL Regular Languages (RL), that is, if L ⊂ R then L is a RL. Similarly, call C the set of ALL Context Free Languages (CFL).

Prove/disprove the following statements:

(i) If L1 ∈ R and L1 ∪ L2 ∈ R then L2 ∈ R.

(That is, you are asked to prove or disprove that if L1 is a regular language and the union L1 ∪ L2 is also a regular language then prove/disprove the statement that L2 is also a regular language).

(ii) If L3 ∈ C and L3 ∩ L4 ∈ C then L4 ∈ R.

[40pts] Answer:

(i) [20pts] (Linz 4.1 #20)

[5 pts] Disprove. L2 is not necessarily regular.

[5 pts] Assume that L1 is \* which of course is regular since  is regular and so is its star set.

[5 pts] Now the union of \* (the universe set of languages!) with any set L2 is \* no matter what L2 may be because L2 is always a subset of .

[5 pts] Then it is absurd that L2 may be regular because we could have select L2 = {anbn: n>0} which is not regular and the union is still regular.

(ii) [20pts] (my invention)

[5 pts] Disprove. L4 is not necessarily regular.

[5 pts] Assume that L3 is  which of course is regular by definition and thus, it is also a CFL, since all regular languages are CFL.

[5 pts] Now the intersection of  with any set L4 is  no matter what L4 may be because the intersection of any set with the empty set is the empty set.

[5 pts] Then it is absurd that L4 may be regular because we could have select L4 = {anbn: n>0} which is not regular and the intersection is still context free (the empty set).

Problem #2

*Design* a PDA for the language L = {an bn cm dm / m, n ≥ 1}. Carefully define all its transitions and identify all its states including the accepting state. Your answer must include a PDA picture annotated with all its corresponding transitions as shown in class for the PDA recognizing wwR. (Do not do more than 5 states!)

Answer:

[25 pts] The best answer could be a picture as follows:

[5 pts]

q4

q1

, a 🡺 a

q0

a, $ 🡺 a$

a, a 🡺 aa

b, a 🡺 

, $ 🡺 $

q2

c, $ 🡺 c$

c, c 🡺 cc

, c 🡺 c

q3

d, c 🡺 

, $ 🡺 $

[5 pts]

[5 pts]

[5 pts]

[5 pts]

[However, your answer could also be a PDA with only two states (this is the minimum number of states. It could go like this:

At the initial state, q0, we have the following transitions:

[3 pts] a, $ 🡺 a$

[3 pts] a, a 🡺 aa

[3 pts] b, a 🡺 

[3 pts] c, $ 🡺 c$

[3 pts] c, c 🡺 cc

[3 pts] d, c 🡺 

[3 pts] , $ 🡺 $ transfer to the [2 pts] accepting/final state, q2.

[2 pts] *In this case, your picture will include only two bullets, the first one and the last one shown above. There are variations in between my best solution and the one just explained …See below.*

[25 pts]

q1

q0

a, $ 🡺 a$

a, a 🡺 aa

b, a 🡺 

c, $ 🡺 c$

c, c 🡺 cc

d, c 🡺 

, $ 🡺 $

Problem #3

Use the pumping lemma (PL) for regular languages to prove that the language following language

L = {anbncm : m, n ≥ 1} is not a Regular Language (RL).

I will consider that, as usual, you are assuming that L is a RL and thus you are looking for a contradiction based on the conditions of the PL (do not to say a word about it, just go to the point). For this, you will consider that *m* *is the pumping length* (Do not use any other letter for pumping length). *Then you first step is to define string s ∈ L (this is worth your first 5 points)*. Next, you will consider only the following TWO cases:

1. y is *only a’s*. That is, let y = ar, with r > 0, so that |y| > 0 (PL’s *second constraint*), and
2. y is *a’s and b’s*. That is, let *y = arbt* with r, t > 0 (to have a’s and b’s in the expression of y, exponents r and t have to be non-zero). Hence, |y| = r + t > 0 (PL’s *second constraint*).

Using all this information you must show *in detail (do all the necessary algebra)* how to reach the desired contradiction (that’s all) making sure that the PL’s *third constraint* is taken care of.

Answer [25pts]:

1. [5pts] Let s = apbpcq = xyz ∈ L, with p, q ≥ 1, and thus we have
2. [5pts] the following *constraint:* |s| = p + p + q = 2p +q *≥*  m
3. [5pts] Case (i): Assume x = ak, with k = p – r, y = ar, and thus z = bpcq (z is evaluated after selecting k + r = p. In this case, the PL’s *third constraint* is |xy| = | akar| = k + r = p ≤ m. This give us the string s = xyz = (ak)( ar)bpc = apbpcq.
4. [5pts] Now we need to have *xyiz ∈ L*, for *i ≥ 0*, that is, this string becomes ak(ar)ibpcq = ak+ribpcq, but since k = p- r we have *xyiz* = ap-r+ribpcq = ap+(r-1)ibpcq (here *i* has to be > 0 for the exponent to be positive and also i has to be ≠ 1, otherwise we end up with the initial s). In summary, for *i ≥ 2*, this term will have too many *a’s* and thus, *xyiz ∉ L*. Contradiction!
5. [5pts] Case (ii): Given *y = arbt*. Consider s = xyz = apbpcq = xarbtz and we can choose x = ap-r and z = bp-tcq. However, it is immediate that for *xyiz ∈ L*, *i ≥ 0* we will have the following *xyiz* = x(*arbt*)iz = ap-r (*artt*)(*arbt*)… (*arbt*)bp-tcq. This expression will have a’s followed by b’s a total of “i” times (in this case the a’s and b’s are intermixed for i >1) plus a different number of *a’s* and *b’s* for i = 0 (since in general r ≠ t) for any fixed choice of x and z (*fixed,* meaning independent of *i*) and thus *xyiz ∉ L*. Contradiction!
6. [0pts] We cannot find a *y* that will satisfy the pumping lemma without reaching a contradiction. Therefore, L is not regular.

Problem #4

Given the CFL L = {wwR: w in {a,b}\*}, use its corresponding non-deterministic PDA to recognize the string *w = aabbbbaa*. That is, use the PDA already studied in class to show ALL the transitions needed to *accept* or *reject* (whatever takes place first!) the string *w* given above

Answer [40 pts]:

We go to class notes 4b page 20 to use the PDA that recognizes strings in wwR. The table in the picture includes 6 activities at q0 and 2 activities at q1 plus 3 transition conditions from q0 to q1 and 1 condition transition from q1 to q2. Just make 0 = a and 1 = b!).

1pt ,$/$ 1pt ,$/$ 1pt

(q0, aabbbbaa, $) 🡺 (q1, aabbbbaa, $) 🡺 (q2, aabbbbaa, $) nafs (not a final state)

↓a,$/a$ (stays at q0 stacking a)

1pt ,a/a 1pt a,a/ 1pt ,$/$ 1pt

(q0, abbbbaa, a$) 🡺 (q1, abbbbaa, a$) 🡺 (q1, bbbbaa, $) 🡺 (q2, bbbbaa, $) nafs

↓ a,a/aa (stays at q0 stacking a)

1pt ,a/a 1pt

(q0, bbbbaa, aa$) 🡺 (q1, bbbbaa, aa$) it is stuck at a nafs (the stack is not empty)

↓ b,a/ba (stays at q0 stacking b)

1pt ,b/b 1pt b,b/ 1pt

(q0, bbbaa, baa$) 🡺 (q1, bbbaa, baa$) 🡺 (q1, bbaa, aa$) it is stuck at a nafs (the stack is not empty)

↓ b,b/bb (stays at q0 stacking b)

1pt ,b/b 1pt b,b/ 1pt b,b/ 1pt a,a/ 1pt a,a/

(q0, bbaa, bbaa$) 🡺 (q1, bbaa, bbaa$) 🡺(q1, baa, baa$) 🡺 (q1, aa, aa$) 🡺(q1, a, a$) 🡺

a,a/,$/$ 1pt

🡺 (q1, , $) 🡺 (q2, , $) accept!

Total 33pts: 19 IDs + 14 transitions (1 pt each).

Problem #5

Given the language L = {anbmcp: m > n + p }

1. *Describe* a PDA to accept this language
2. Write a grammar to represent this language, Explain, in detail, how you came up with the grammar you are using! Justify your grammar in detail!

Answer:

(i)[16 pts]

1. [4pts] The a’s are coming and we push them into the stack.

2. [4pts] When the b’s come, we pop the stack until it is empty (m > n). We push the remaining b’s into the stack.

3. [4pts] When the c’s are coming we pop the stack.

4. [4pts] When we run out of c’s, we accept if there are b’s in the stack (m > n + p), otherwise we reject.

(ii)[25pts]

1. [5pts] Just write L as L = {anbnbqbpcp: m = n + q + p; n, p ≥ 0 and q > 0}and thus,
2. [15pts] L is the concatenation of CFLs,

[5pts] L1 = {anbn: n ≥ 0}; [5pts] L2 = {bq: q > 0} and [5pts] L3 = {bpcp: p ≥ 0} with the following corresponding grammars

1. [3pts] [1pt] S1 🡺aS1b | ; [1pt] S2 🡺 bS2 | b; [1pt] S3 🡺 bS3c | ;
2. [2pts] Lastly: S 🡺S1S2S3 is the requested grammar.

Problem #6

Use the pumping lemma (PL) to prove that the language L = {a3n#b5n#c7n, n ≥ 0} is not a Context Free (CF).

I will consider that, as usual, you are assuming that L is a CF Language (CFL) and thus you are looking for a contradiction based on the conditions of the PL for CFLs. For this, you will consider that *m* is the pumping length. *Then your first step will be to define string s ∈ L and its constraint*. Next, you shall consider only the following THREE cases:

1. Discuss what can happen if v or y include #.
2. v is only *a’s*
3. y is only c’s

Using all this information you must describe why we reach a contradiction in each case. (Notice that the length ratios are |a|:|b|:|c| = ???). *Do not to do all the algebra, just give a complete explanation, that should suffice.*

[20 pts] Answer:

(i)[5pts] Let s = a3p#b5p#c7p, with the constraint |s| = 3p +1 + 5p +1 + 7p = 15p + 2 ≥ m, p ≥ (m-2)/15

(ii)[5pts] Neither v nor y can contain # because uv2xy2z (case of i ≥ 2) will contain more than two #s

(iii)[5pts] If v is only a’s then uv2xy2z will contain extra a’s and the ratio 3:5:7 will not be maintained. The 3 will be increased. (The corresponding Algebra can be done by putting v = ar and thus s = uvxyz = u(ar)xyz = a3p#b5p#c7p 🡺 A possible selection is: u = a3p-r, x = #, y =  and z = b5p#c7p and the condition |vxy| = r + 1 + 0 = r + 1 ≤ m, that is: r ≤ m – 1).

(iv)[5pts] If y is only c’s then uv2xy2z will contain extra c’s and the ratio 3:5:7 will not be maintained. The 7 will be increased. (The algebra is similar to the one above)