**Linguagens Formais e Autômatos**

Arthur do Prado Labaki – 11821BCC017

**Segunda Lista de Exercícios**

**Parte 1**

1) **Lema do Bombeamento para linguagens regulares:**

Prove que as linguagens a seguir não são regulares

**L1** = {w ϵ {a, b}\* | w = w1w1R }. Ex: abbbba.

Assumindo que L1 é regular, o lema do bombeamento é valido.

Sendo p o valor dado pelo lema e w = apbbap.

Como |w| ≥ p, w pode ser escrito como uvz.

Qualquer que seja a divisão em que v ≥ 1 e |uv| ≤ p, temos v = at, para algum t ≥ 1 e u = ar, para algum r ≥ 0.

Então uvvz = aratatap-r-tbbap. Mas r+2t+p-r-t = p+t e não há forma de dividir ap+tbbap em duas partes (wwr).

Logo uvvz não pertence a L1, uma contradição do lema. Então L1 não é uma linguagem regular.

**L2** = {w ϵ {0, 1}\* | w = 0i1j, sendo i > j, j ≥ 0}. Ex: 0001.

Assumindo que L2 é regular, o lema do bombeamento é valido.

Sendo p o valor dado pelo lema e w = 0p+11pe p ≥ 0.

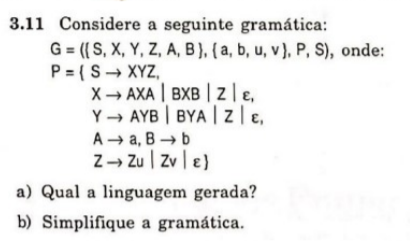
Como |w| ≥ p, w pode ser escrito como uvz.

Qualquer que seja a divisão em que v ≥ 1 e |uv| ≤ p, temos v = 0t, para algum t ≥ 1 e u = 0r, para algum r ≥ 0.

Então uz = 0r0p+1-r-t1p. Mas r+p+1-r-t = p+1-t ≤ p, não obedecendo a regra i > j.

Logo uz não pertence a L2, uma contradição do lema. Então L2 não é uma linguagem regular.

2) **Simplificação e normalização de Gramáticas Livres de Contexto**



a) A Linguagem gerada é a junção de duas linguagens conhecidas, a wcw e a wcwr, como também suas combinações.

b) Utilizando simplificações combinadas temos:

Retirada de produções vazias:

G = ({S, X, Y, Z, A, B}, {a, b, u, v}, P, S), onde:

P = {S → XYZ | YZ | XZ | XY | X | Y | Z | ε,

X → AXA | BXB | Z | AA | BB, Y → AYB | BYA | Z | AB | BA,

Z → Zu | Zv | u | v, A → a, B → b}

Retirada de produções que substituem variáveis:

G = ({S, X, Y, Z, A, B}, {a, b, u, v}, P, S), onde:

P = {S → XYZ | YZ | XZ | XY | ε | AXA |BXB | AA | BB | AYB | BYA | AB | BA | Zu | Zv | u | v,

X → AXA | BXB | AA | BB | Zu | Zv | u | v,

Y → AYB | BYA | AB | BA| Zu | Zv | u | v,

Z → Zu | Zv | u | v, A → a, B → b}

Retirada de símbolos inúteis:

G = ({S, X, Y, Z, A, B}, {a, b, u, v}, P, S), onde:

P = {S → XYZ | YZ | XZ | XY | ε | AXA |BXB | AA | BB | AYB | BYA | AB | BA | Zu | Zv | u | v,

X → AXA | BXB | AA | BB | Zu | Zv | u | v,

Y → AYB | BYA | AB | BA| Zu | Zv | u | v,

Z → Zu | Zv | u | v, A → a, B → b}

PS: Não encontrei casos em que substituía os estados A e B, mesmo eles levando a somente uma única variável (Como AXA iria ser aXa, retirando esses estados).

3) **Ambiguidade**

Complemente a gramática de expressões vista em aula incluindo as operações de divisão (/) e subtração (-), além de 2 operandos (y e z):

G = ({E}, {+, -, \*, /, [, ], x, y, z), P, E),

P = {E → E+E | E\*E | E-E | E/E | [E] | x | y | z }

Remova a ambiguidade dessa gramática, considerando que + e – têm a mesma precedência, que é menor que a precedência de \* e /, sendo que esses 2 últimos têm a mesma precedência.

Mostre todas as árvores de derivação possíveis nas duas gramáticas (ambígua e não ambígua) para a expressão:

x / [y + z \* x] – y

Removendo a ambiguidade da gramatica utilizando as regras de precedências, temos:

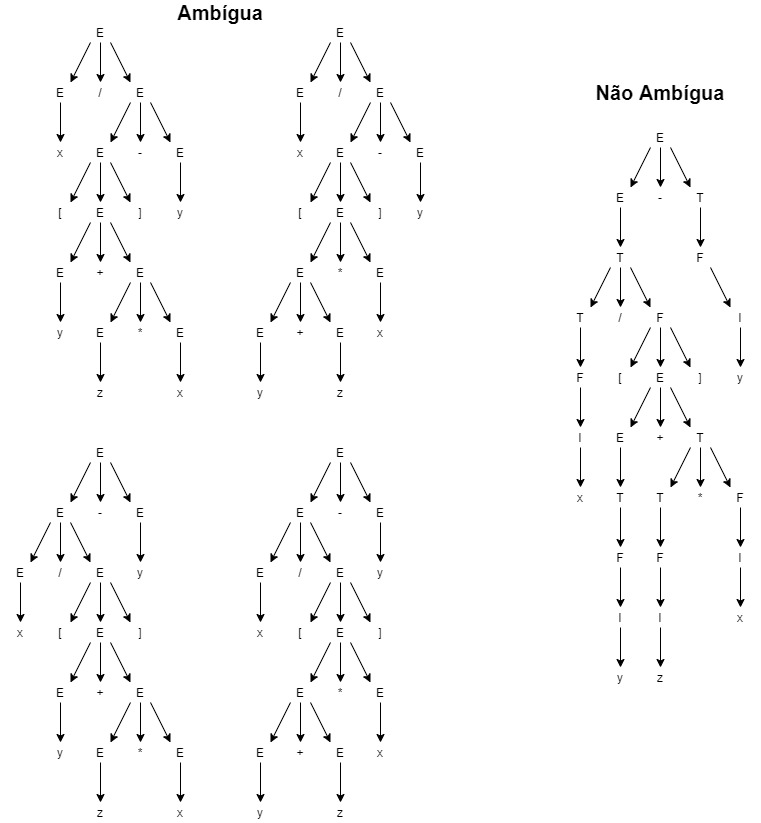
G2 = ({E, T, F, I}, {+, -, \*, /, [, ], x, y, z), P2, E), onde

P2 = { E → T | E + T | E – T,

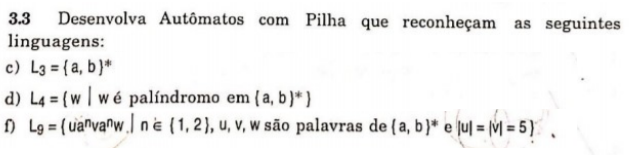
T → F | T \* F | T / F,

F → I | [ E ],

I → x | y | z }



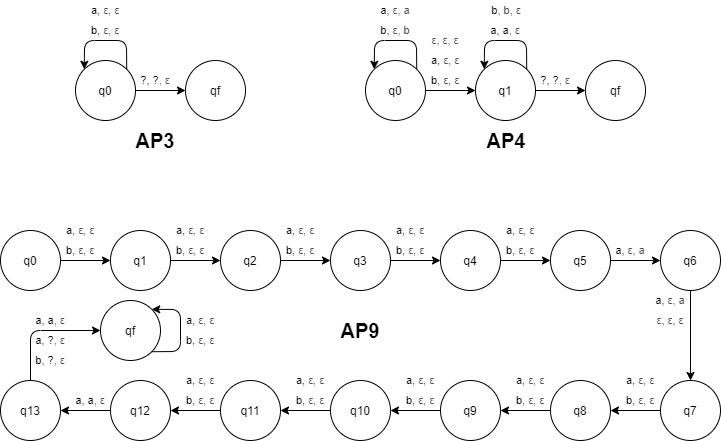
4) **Autômato de Pilha**



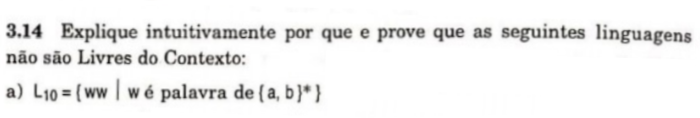
AP3 = ({ a, b }, { q0, qf }, δ1, q0, { qf }, { a, b })

AP4 = ({ a, b }, { q0, q1, qf }, δ2, q0, { qf }, { a, b })

AP9 = ({ a, b }, { q0, q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12, q13, qf }, δ3, q0, { qf }, { a, b })



5) **Lema do Bombeamento para Linguagens Livres de Contexto**



Supondo que L10 é uma LLC e ‘n’ o comprimento do bombeamento, vamos considerar w = anbnanbn, com |w|≥ p.

Pelo lema do bombeamento, w = uxvyz, tal que:

|xvy| ≤ n e |xy| ≥ 1 para todo i ≥ 0, sendo uxivyix

Assim, supondo que x e y contém somente símbolos a ou somente símbolos b (4 casos)

Nesse caso, ux2vy2x contém mais símbolos que a cadeia original, logo é uma contradição do lema. No caso a ou b seria maior que n, por exemplo, an+vbnanbn

Supondo que x e y contém parte de símbolos a e b (3 casos)

Ainda assim, ux2vy2x contém mais símbolos que o resto da cadeia. No caso, algum anbn seria sempre maior que o resto da palavra, não respeitando sua regra.

Logo uxvyz não pertence a L10, uma contradição do lema. Então L10 não é uma linguagem livre de contexto.