EXERCÍCIO 1

GRAMÁTICA G
P = { S → 0S0 | c0c0c,
S → 1S1 | c1c1c,
S → c }

CRIAÇÃO DE UM NOVO SÍMBOLO NÃO TERMINAL INICIAL

P0 → P

ELIMINAÇÃO DAS PRODUÇÕES VAZIAS

A GRAMÁTICA NÃO POSSUI PRODUÇÕES VAZIAS, NÃO POSSUI λ.

ELIMINAÇÃO DAS PRODUÇÕES UNITÁRIAS

(Gramática atual)

P0 → P
P = { S → 0S0 | c0c0c,
S → 1S1 | c1c1c,
S → c }

(Gramática Atualizada com base na regra, P0 passa a produzir tudo o que S produz com base na regra)

P0 → 0S0 | c0c0c,
S → 1S1 | c1c1c,
S → c
P = { S → 0S0 | c0c0c,
S → 1S1 | c1c1c,
S → c }

ELIMINAÇÃO DOS SÍMBOLOS INÚTEIS OU INACESSÍVEIS

TODOS OS SÍMBOLOS DA GRAMÁTICA PODEM SER ACESSADOS.

PADRONIZAÇÃO REGRAS X → YX | X → x | X → λ

(Gramática atual, antes da padronização)
P0 → 0S0 | c0c0c,
S → 1S1 | c1c1c,
S → c
P = { S → 0S0 | c0c0c,
S → 1S1 | c1c1c,
S → c }

(Atualização)

$X^1 \rightarrow 0S$
$X^2 \rightarrow S1$
$X^3 \rightarrow c1c$
$Y^1 \rightarrow c0c$
$Y^2 \rightarrow 0c$
$Y^3 \rightarrow 1c$

$P0 \rightarrow X^10 \mid Y^1Y^2$,
$S \rightarrow X^21 \mid X^3Y^3$,
$S \rightarrow c$
$P = \{ S \rightarrow X^10 \mid Y^1Y^2$,
$S \rightarrow 1X^2 \mid X^3Y^3$,
$S \rightarrow c \}$

GRAMÁTICA RESULTANTE = ({P0,S,X¹,X²,X³,Y¹,Y²,Y³,c,1})

$X^1 \rightarrow 0S$
$X^2 \rightarrow S1$
$X^3 \rightarrow c1c$
$Y^1 \rightarrow c0c$
$Y^2 \rightarrow 0c$
$Y^3 \rightarrow 1c$

$P0 \rightarrow X^10 \mid Y^1Y^2$,
$S \rightarrow X^21 \mid X^3Y^3$,
$S \rightarrow c$
$P = \{ S \rightarrow X^10 \mid Y^1Y^2$,
$S \rightarrow 1X^2 \mid X^3Y^3$,
$S \rightarrow c \}$

Questão B = pode ser utilizada para a produção da linguagem L w Є {0, 1, c} pois a linguagem L wcw é elevado a R que pode ser de qualquer tamanho indicando que podem haver distribuições de símbolos diferentes mesmo w pertencendo a linguagem {0, 1, c}.

EXERCÍCIO 4 PARTE 1, O DIAGRAMA ESTÁ NA FOTO ABAIXO

ETAPAS DA APLICAÇÃO DO LEMA DO BOMBEAMENTO PARA LINGUAGENS REGULARES

1) ASSUMIR QUE A LINGUAGEM É REGULAR

L1 = { 01*1 } É REGULAR.

2) DEFINIR UM NÚMERO INTEIRO p (COMPRIMENTO DO BOMBEAMENTO) SENDO QUE p>= 1 e p <= |w| SENDO QUE w É UMA PALAVRA/CADEIA/STRING/SENTENÇA DE L

PALAVRA/CADEIA/STRING/SENTENÇA w = 0011

representando o fecho de Kleene Com k = 1 e t = 3, tem-se 01 elevado a t 1, e p = (k + t), isto é, p = Assim p contempla a primeira parte a esquerda antes da primeira parte bombeável (símbolo 0) e a primeira parte bombeável (símbolos 111).
Escolhi para o exercício: p = 3

Agora seguindo o lema preciso encontrar uma palavra/cadeia/string/sentença válida para a linguagem L. Palavras/cadeias/strings/sentenças válidas para a linguagem L: 01, 011, 01111

Escolha para o exercício: 01111, que é a palavra/cadeia/string/sentença w

3) Definir xyz, sendo que x, y e z representam subpalavras/subcadeias da palavra/cadeia/string/sentença escolhida. O objetivo é que x contemple a primeira parte da palavra/cadeia/string/sentença, a esquerda e antes da primeira parte bombeável; y a primeira parte bombeável e z o restante da palavra/subcadeia.
As subpalavras/subcadeias x e z podem ser vazias, isto é, com comprimentos iguais a zero, mas o comprimento de y deve ser maior ou igual a 1.
Considerando o exercício com a string 01111:
x = 0
y = 11
z = 11

4) Encontrar palavras/cadeias/strings/sentenças a partir de xyi z, considerando i >= 0.
Observação: o y representa uma palavra/cadeia/string/sentença 01, então y elevado a 0 = λ, y^1 = 01, y^2 = 0111.

Escolhas para o exercício:
Se i = 0, xy elevado a i z, tem-se 011
Se i = 1, xy elevado a i z, tem-se 01111
Se i = 2, xy elevado a i z, tem-se 011111
Se i = 3, xy elevado a i z, tem-se 011111111

5) Verificar se as palavras/cadeias/strings/sentenças geradas na etapa 3 pertencem a linguagem L = {01*1}, similar a palavra/cadeia/string/sentença da etapa 2. Se sim, a linguagem é regular, senão, por contradição a linguagem não é regular.
Com base no lema do bombeamento: as palavras/cadeias/strings/sentenças pertencem a L, então a linguagem L é regular porque elas começam com 0 e terminam com 1 podendo haver vários outros 1 no meio de 0 e 1.

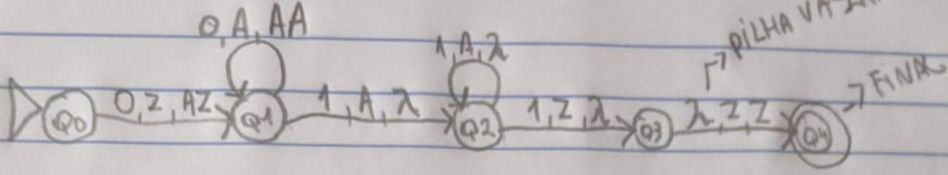
VERIFICAÇÃO DE EQUIVALÊNCIA ENTRE AS LINGUAGENS L1 E L2

L1 = { 01*1 } <=> L2 = { (0 + 1)* } não são equivalentes porque não podem gerar as mesmas palavras um exemplo seria

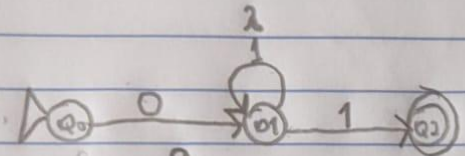
L1 = 01 & L2 = poderia ser vazio porque é elevado a feche de kleene e como esta (0 + 1)* poderia gerar lambda já L1, Não pode ser vazio pois apenas o 1 é elevado a feche de kleene

RESOLUÇÃO DOS OUTROS EXERCÍCIOS

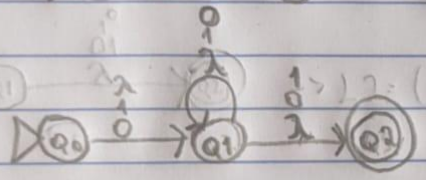
EX 03)



EX 04)



→ L1 = 01*1



L2 = (0+1)*

EX 05)