Linguagens Formais e Autômatos

Arthur do Prado Labaki – 11821BCC017

Primeira Lista de Exercícios

1) Sejam as gramáticas G1, G3 e G5 abaixo.

a) Descreva qual é a linguagem gerada por cada gramática abaixo.

b) Apresente qual é o tipo (0, 1, 2 ou 3) de cada gramática? Justifique.

G1 = ({S,A,B,C,E,F}, {0,1,2}, {S→012|0BAC, B→0B1A|01A, AC→C2,1C2→F12, AF→FA, FA1→F1E, E2→22, 1F1→F11, 1E1→11E, 0F1→011}, S)

Exemplos:

$S \Rightarrow 012$

$S \Rightarrow 0\underline{B}AC \Rightarrow 001A\underline{AC} \Rightarrow 001\underline{AC}2 \Rightarrow 00\underline{1C2}2 \Rightarrow 0\underline{0F1}22 \Rightarrow 001122$

$S \Rightarrow 0\underline{B}AC \Rightarrow 00\underline{B}1AAC \Rightarrow 0001A1A\underline{AC} \Rightarrow 0001A1\underline{AC}2 \Rightarrow 0001A\underline{1C2}2 \Rightarrow 0001\underline{AF}122 \Rightarrow 0001\underline{FA1}22 \Rightarrow 0001\underline{F1E}22 \Rightarrow 000\underline{1F1}222 \Rightarrow 00\underline{0F1}1222 \Rightarrow 000111222$

$S \Rightarrow 0\underline{B}AC \Rightarrow 00\underline{B}1AAC \Rightarrow 000\underline{B}1A1AAC \Rightarrow 00001A1A1A\underline{AC} \Rightarrow 00001A1A1\underline{AC}2 \Rightarrow 00001A1A\underline{1C2}2 \Rightarrow 00001A1\underline{AF}122 \Rightarrow 00001A1\underline{FA1}22 \Rightarrow 00001A1F1\underline{E2}2 \Rightarrow 00001A\underline{1F1}222 \Rightarrow 00001\underline{AF}11222 \Rightarrow 00001\underline{FA1}1222 \Rightarrow 00001F\underline{1E1}222 \Rightarrow 00001F11\underline{E2}22 \Rightarrow 0000\underline{1F1}12222 \Rightarrow 000\underline{0F1}112222 \Rightarrow 000011112222$

Logo a linguagem gerada é:

L(G1) = {w ϵ {0,1,2}* | w = $0^n1^n2^n$, N>0}

Não é gramatica do tipo 3 (regular), pois tem mais de uma variável a direita (B→0B1A)

Não é do tipo 2 (livre de contexto), pois tem mais de uma variável a esquerda (AF→FA)

É gramatica do tipo 1 (sensíveis ao contexto), pois sendo $\alpha \rightarrow \beta$, $|\beta| \geq |\alpha|$

G3 = ({S, A, B, C, D, E}, {0}, P3, S),

P3 = {S→AC0B, C0→000C, CB→DB, CB→E, 0D→D0, AD→AC, 0E→E0, AE→ε}

Exemplos:

$S \Rightarrow A\underline{C0}B \Rightarrow A000\underline{CB} \Rightarrow A00\underline{0E} \Rightarrow A0\underline{0E}0 \Rightarrow A\underline{0E}00 \Rightarrow \underline{AE}000 \Rightarrow 000$

$S \Rightarrow A\underline{C0}B \Rightarrow A000\underline{CB} \Rightarrow A00\underline{0D}B \Rightarrow A0\underline{0D}0B \Rightarrow A\underline{0D}00B \Rightarrow \underline{AD}000B \Rightarrow A\underline{C0}00B \Rightarrow A000\underline{C0}0B \Rightarrow A000000\underline{C0}B \Rightarrow A000000000\underline{CB} \Rightarrow A00000000\underline{0E} \Rightarrow A0000000\underline{0E}0 \Rightarrow A000000\underline{0E}00 \Rightarrow A00000\underline{0E}000 \Rightarrow A0000\underline{0E}0000 \Rightarrow A000\underline{0E}00000 \Rightarrow A00\underline{0E}000000 \Rightarrow A0\underline{0E}0000000 \Rightarrow A\underline{0E}00000000 \Rightarrow \underline{AE}000000000 \Rightarrow 000000000$

Logo a linguagem gerada é:

L(G3) = {w ϵ {0}* | w = $0^{3\text{^}N}$ (0 elevado ao cubo, elevado à N), sendo N ≥ 1}

Não é gramatica do tipo 3 (regular), pois tem mais de uma variável a direita (CB→DB)

Não é do tipo 2 (livre de contexto), pois tem mais de uma variável a esquerda (AD→AC)

Não é do tipo 1 (sensíveis ao contexto), pois o lado esquerdo é maior que o direito (AE→ε)

Logo é do tipo 0 (irrestrita), pois tanto o lado direito quanto esquerdo ϵ (V U T) *

G5 = ({E,O}, {+,*, id}, P5, E),

P5 = {E → EOE , E → id, O → +, O → *}

Exemplos:

$E \Rightarrow id$

$E \Rightarrow \underline{E}OE \Rightarrow idO\underline{E} \Rightarrow id\underline{O}id \Rightarrow id + id$

$E \Rightarrow \underline{E}OE \Rightarrow idO\underline{E} \Rightarrow id\underline{O}id \Rightarrow id * id$

$E \Rightarrow \underline{E}OE \Rightarrow idO\underline{E} \Rightarrow idO\underline{E}OE \Rightarrow idOidO\underline{E} \Rightarrow id\underline{O}idOid \Rightarrow id{+}id\underline{O}id \Rightarrow id{+}id{*}id$

$E \Rightarrow \underline{E}OE \Rightarrow idO\underline{E} \Rightarrow idO\underline{E}OE \Rightarrow idOidO\underline{E} \Rightarrow id\underline{O}idOid \Rightarrow id{*}id\underline{O}id \Rightarrow id{*}id{*}id$

$E \Rightarrow \underline{E}OE \Rightarrow idO\underline{E} \Rightarrow idO\underline{E}OE \Rightarrow idOidO\underline{E} \Rightarrow idOidO\underline{E}OE \Rightarrow idOidOidO\underline{E} \Rightarrow id\underline{O}idOidOid \Rightarrow id{*}id\underline{O}idOid \Rightarrow id{*}id{+}id\underline{O}id \Rightarrow id{*}id{+}id{+}id$

Logo a linguagem gerada é:

L(G5) = {w | w representa todas as possíveis expressões de soma e multiplicação, com argumentos id}

Não é gramatica do tipo 3 (regular), pois tem mais de uma variável a direita (E → EOE)

É do tipo 2 (livre de contexto), pois sempre tem uma única variável à esquerda.

2) Construa uma gramática que gere a linguagem:

a) $\Sigma = \{a\}$, $L = \{w \in \Sigma^* \mid$ tamanho de w é múltiplo de 3$\}$

L1 = ({S, B}, {a}, P1, S),

P1 = {$S \rightarrow aaaB$, $B \rightarrow aaaB \mid \varepsilon$}

Assim, sempre começara com o mínimo múltiplo de 3 (o próprio 3) e pode ir aumentando mais 3 (podendo ser aaa, aaaaaa, aaaaaaaaa, aaaaaaaaaaaa, ...).

b) $\Sigma = \{a,b\}$, $L = \{w \in \Sigma^* \mid$ ab é prefixo de w, aa é subpalavra de w e bb é sufixo de w$\}$

L2 = ({S, C}, {a, b}, P2, S),

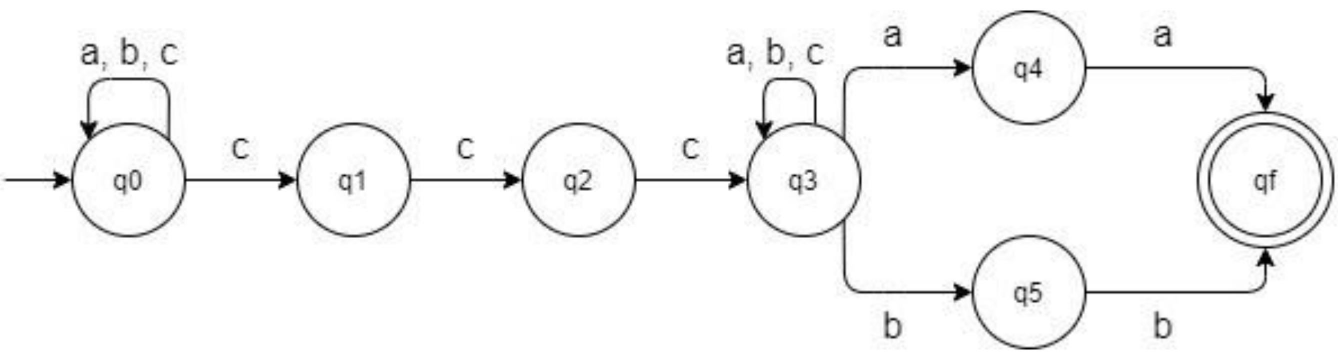
P2 = {$S \rightarrow abCbb$, $C \rightarrow aa \mid aC \mid Ca \mid bC \mid Cb$}

Assim, sempre terá 'ab' como prefixo e 'bb' como sufixo, e pode ter infinitos 'a' ou 'b' dentro da palavra, mas sempre contem a subpalavra 'aa'.

3) Construa um AF que reconheça as linguagens definidas:

a) $\Sigma = \{a,b,c\}$, $L = \{w \in \Sigma^* \mid$ ccc é subpalavra de w e aa ou bb são sufixos de w$\}$

A1 = ({q0, q1, q2, q3, q4, q5, qf}, {a, b, c}, δ, q0, {qf}) sendo δ representado pelas transições do autômato:



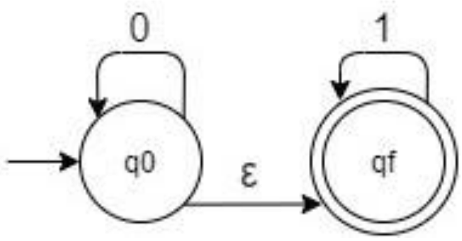
b) $\Sigma = \{0,1\}$, $L = \{w \in \Sigma^* \mid$ qualquer 0 antecede qualquer 1 em $w\}$

A2 = (Q, Σ, δ, q0, F)

Q = {q0, qf}

Σ = {0, 1}

F = {qf}



4) Construa um AF que aceite a linguagem gerada pela gramática

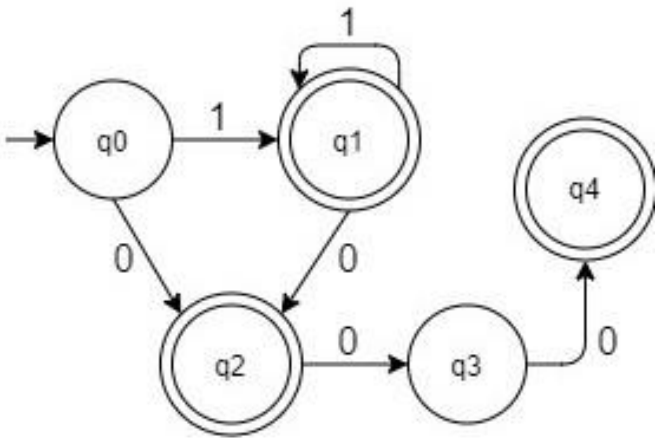
G1 = ({S, A, B}, {0, 1}, {$S \rightarrow 1 \mid 0A \mid 1S \mid 10$, $A \rightarrow \varepsilon \mid 00$}, S)

A1 = (Q, Σ, δ, q0, F)

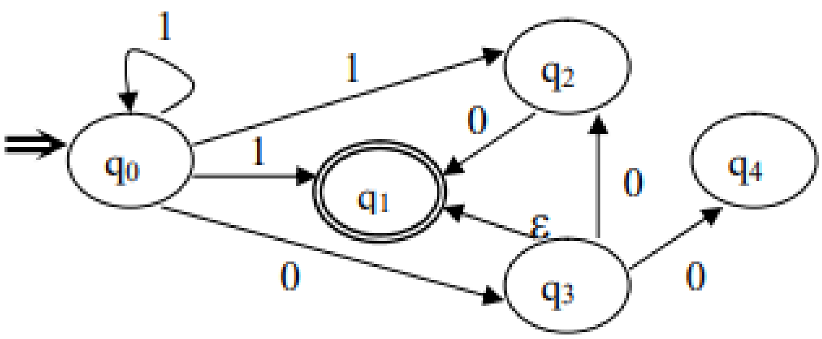
Q = {q0, q1, q2, q3, q4}

Σ = {0, 1}

F = {q1, q2, q4}



5) Construa uma gramática que gere a linguagem aceita pelo AF:



Exemplos Aceitos:

1, 11, 1111111..., 10, 1111111...10, 0, 000.

Logo, sendo G1 = ({S, A, B}, {0, 1}, P1, S):

P1 = {S $\rightarrow$ 1A | 0B, A $\rightarrow$ ε | 1A | 0, B $\rightarrow$ ε | 00}

Aceitando 1^n0^m, sendo N ≥ 0, M = 0, 1 ou 3.

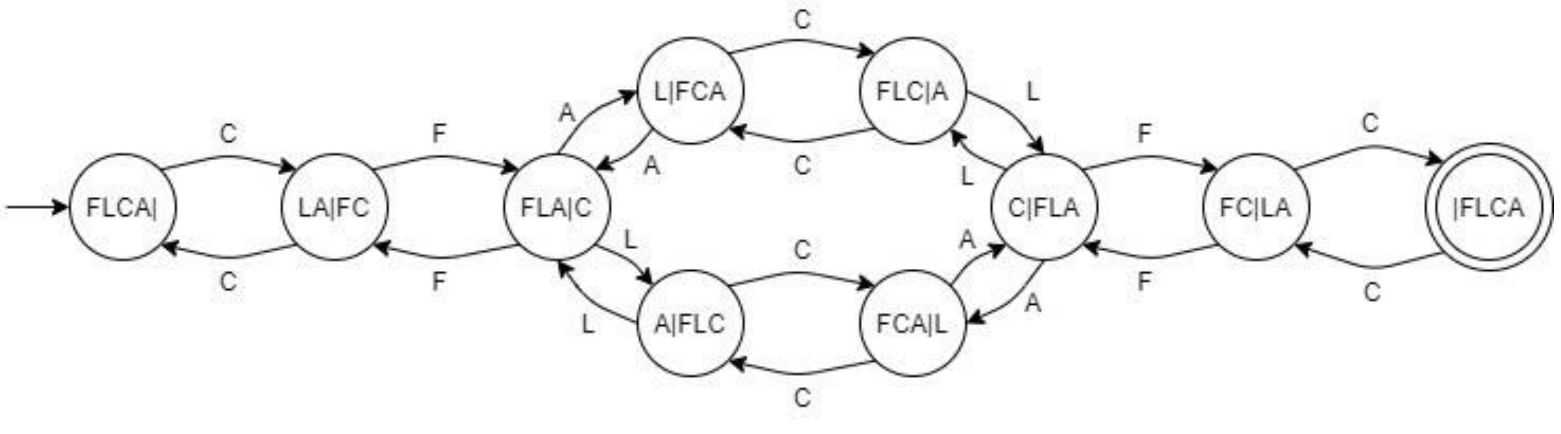
6) Seja o Problema do Fazendeiro

a) Modele esse problema através de um AF, sendo que as transições representam as possíveis travessias de uma margem a outra e os estados as situações válidas possíveis de ocorrer dependendo da transição / travessia utilizada.

Utilizando os estados XXXX|XXXX, sendo antes do rio | depois do rio e F, L, C, A para Fazendeiro, Lobo, Carneiro e Alface respectivamente, temos o conjunto de estados:

Q = {FLCA|, LA|FC, FLA|C, L|FCA, A|FLC, FLC|A, FCA|L, C|FLA, FC|LA, |FLCA}

E seu alfabeto seria $\sum$ = {F, L, C, A} sendo F para somente o fazendeiro, L para fazendeiro e lobo, C para fazendeiro e carneiro e A para fazendeiro e alface.

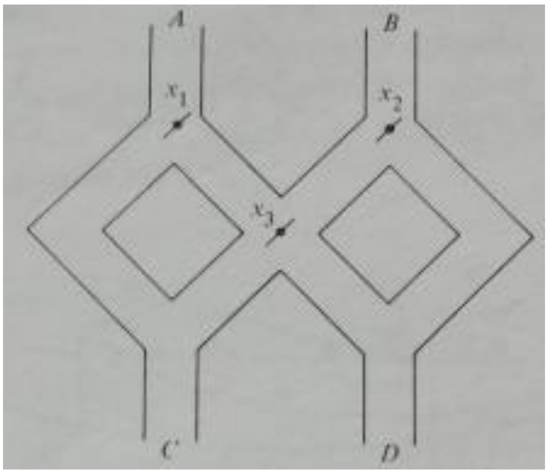


b) Quais são as menores palavras aceitas pelo AF? Ou seja, as sequências mais curtas de travessias?

As menores palavras seriam a que transicionariam diretamente para o estado final, existindo duas menores palavras possíveis, CFACLFC ou CFLCAFC.

Exercícios do livro Hopcroft e Ullman

2.3) Considere o brinquedo apresentado na figura



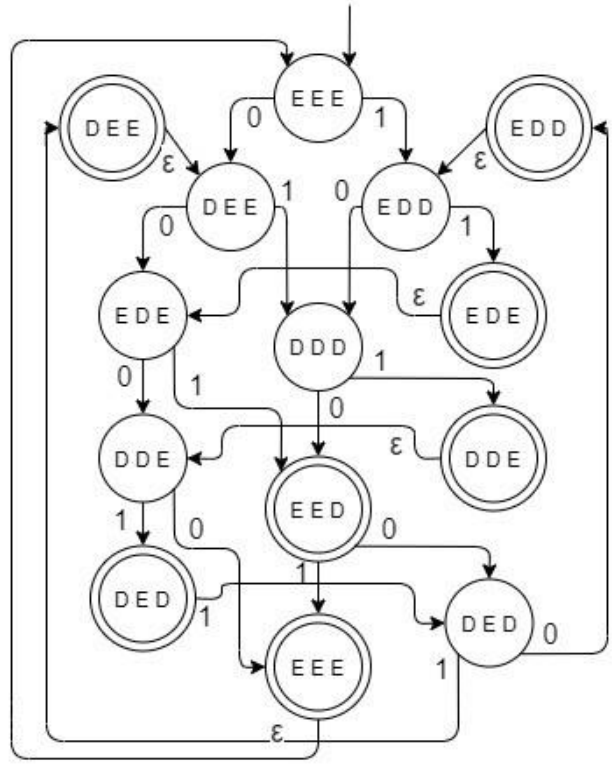
Modele o comportamento desse brinquedo por um autômato finito.

Listando todos os estados, da forma (X1 X3 X2), temos 8 diferentes estados, sendo E para esquerda e D para direita.

Caso algum estado se repita, verificamos se ambos são estados finais ou não.

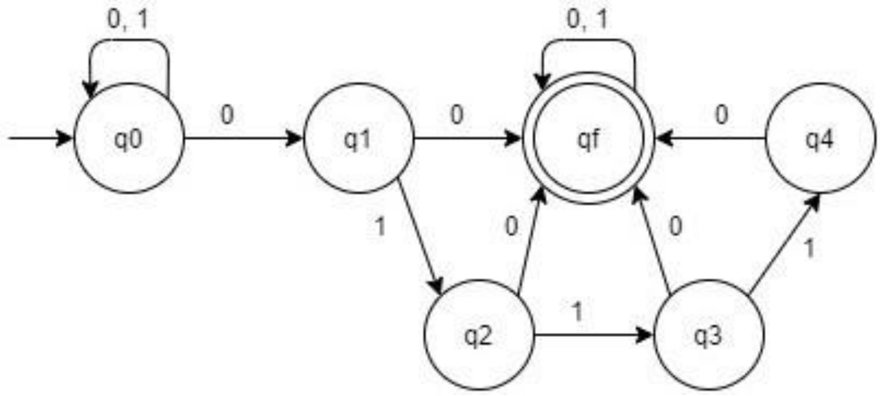
Caso sejam iguais, juntamos em um único estado, para diminuir o AF. Mas caso não sejam, transicionamos o estado final para o mesmo estado não final com uma cadeia vazia.

Supomos que o brbinquedo comece como na imagem (E E E).

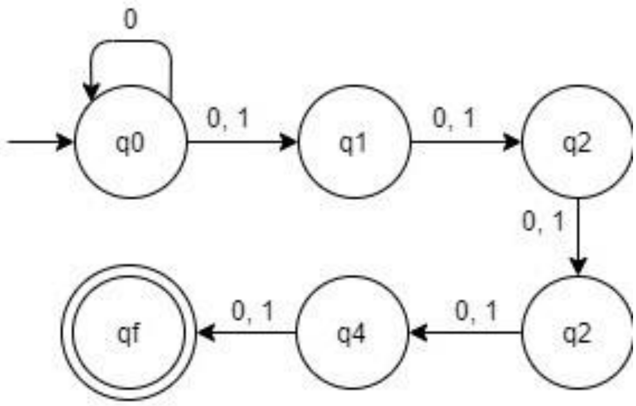


2.5) Escreva um AF que aceita as linguagens a seguir, escritas sobre o alfabeto {0,1}:

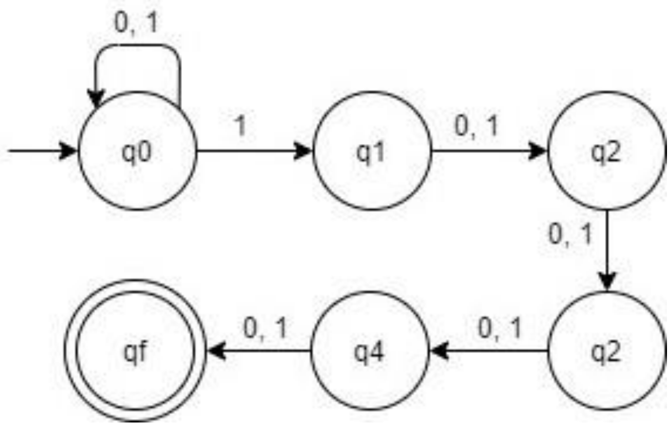
a) Todas as palavras com tamanho maior ou igual a 5 e que para qualquer bloco com 5 símbolos consecutivos da palavra, contém pelo menos 2 zeros.



b) Todas as palavras que podem ser interpretadas como a representação binária de um número inteiro que é congruente a “zero módulo 5”.

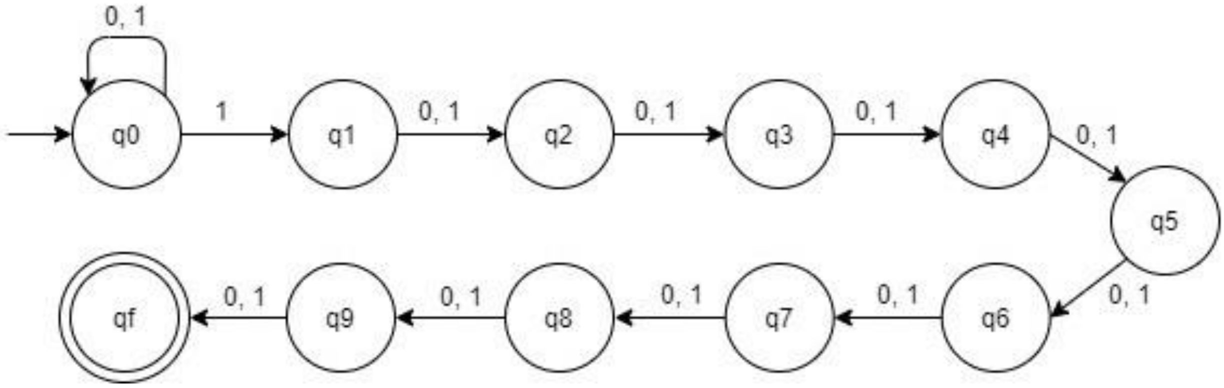


c) Todas as palavras que o quinto símbolo lido da direita para a esquerda é um “1”.



2.8) Escreva um AFND que aceita a linguagem, escrita sobre o alfabeto {0,1}:

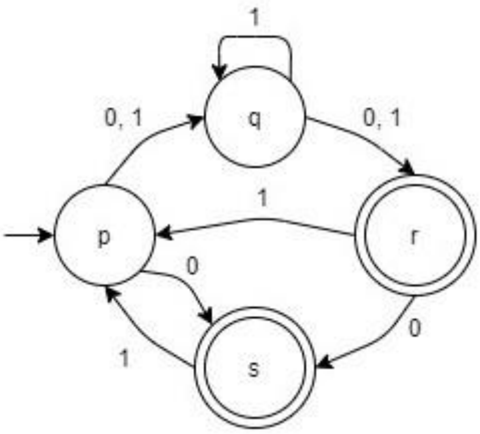
Todas as palavras que o décimo símbolo lido da direita para a esquerda é um “1”.



2.9) Construa o AFD equivalente ao AFD definido pela tabela de transição:

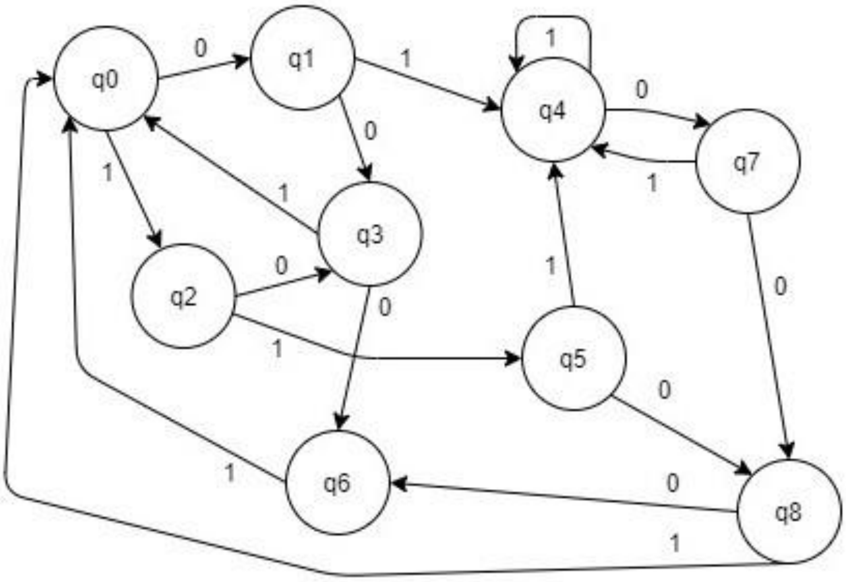
	0	1
⇨ p	q, s	q
q	r	q, r
☼ r	s	p
☼ s	—	p

δ_2



Eu acredito que era para converter essa tabela para um AFD, pois ela é AFN e não AFD como descrito no exercício, logo:

	Q'	0	1
q0	p	qs	q
q1	qs	r	pqr
q2	q	r	qr
q3	r	s	p
q4	pqr	qrs	pqr
q5	qr	rs	pqr
q6	s	-	p
q7	qrs	rs	pqr
q8	rs	s	p

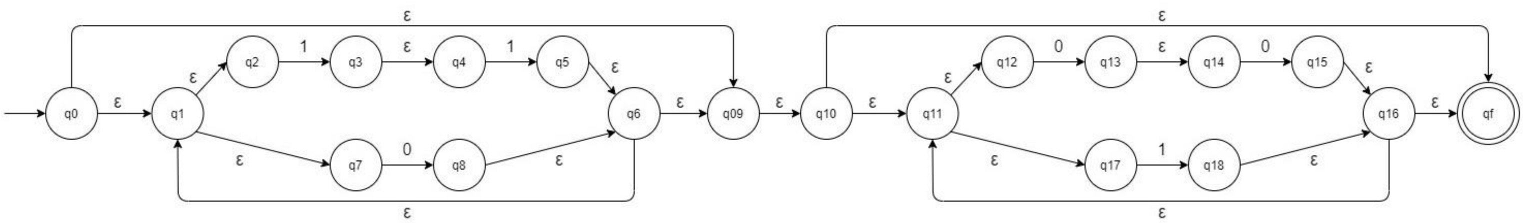


2.11 e 2.12) Descreva as linguagens denotadas pelas expressões regulares e escreva os AFs equivalentes às expressões:

1) (11+0)*(00+1)*

L1 = {w | w tenha depois de uma sequência ímpar de 1s, tenha uma sequência par de 0s}, como:

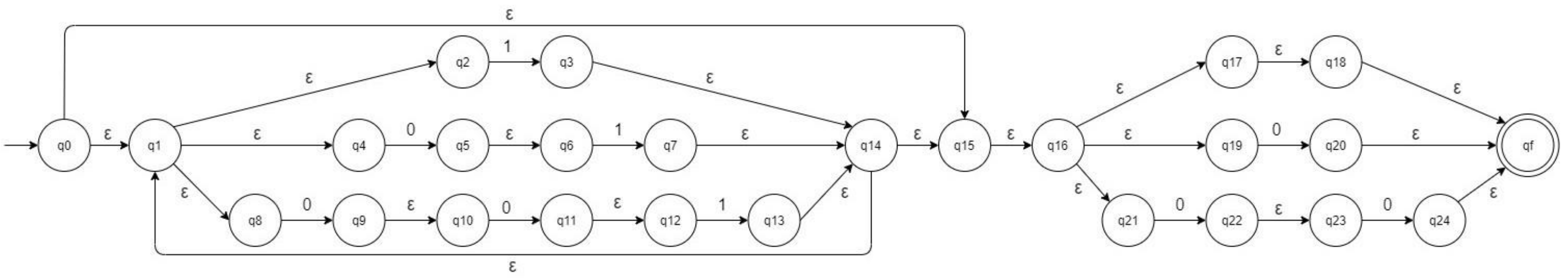
ξ, 0, 1, 00, 01, 11, 000, 10000, 11001100, 1111111111



2) (1+01+001)*(ξ+0+00)

L2 = {w | w tenha no máximo 2 zeros (0) em sequência}, como:

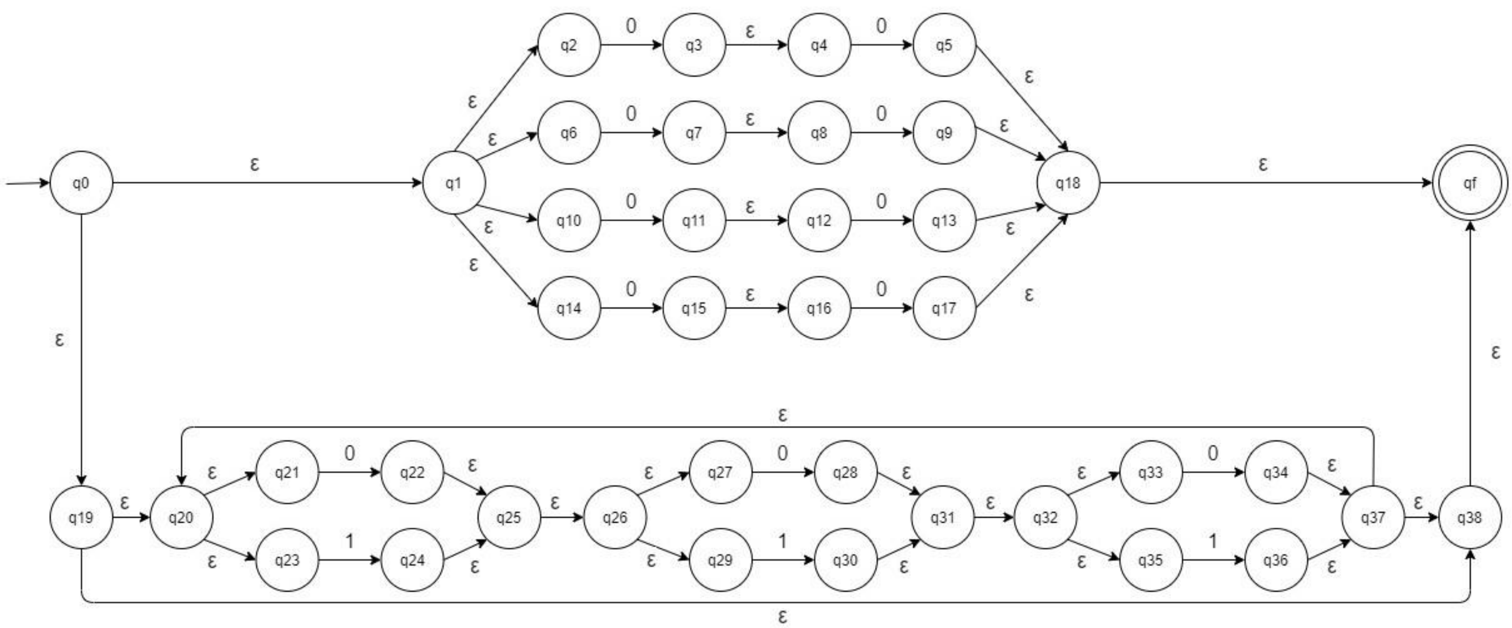
ξ, 0, 1, 00, 01, 11, 111, 0010010010100, 111111, 1001001, 00100



3) (00+01+10+11)* + ((0+1)(0+1)(0+1))*

L3 = {w | w tenha $2^n + 3^m$ de tamanho, sendo N e M ≥ 0}, como:

ξ, 00, 11, 0000, 101010, 100110011010,100101010000000



OBS: Fiz todas as passagens de acordo com o algoritmo de formalismo descrito nos slides da aula, por esse motivo que os autômatos ficaram com muitas transições ε.

2.24 Optei por não encaminhar esse exercício (não encontrei)

4.1 Escreva as Gramáticas (GLC) que geram as linguagens:

a) O conjunto de todas as palíndromos escritas sobre o alfabeto {a,b}

G1 = ({S, A, B}, {a, b}, P, S) onde P é

$S \to aA \mid bB \mid a \mid b \mid \varepsilon$

$A \to Sa$

$B \to Sb$

Exemplos: aa, abba, baba, abbaabba.

b) O conjunto de todas as palavras escritas sobre o alfabeto {(,)}, que representam expressões de parênteses balanceados, isto é, para todo "(" existe um ")", que casam entre si, e todo par "casado" de parênteses está propriamente aninhado.

G2 = ({S, A}, {'(', ')'}, P, S) onde P é

$S \to (A \mid \varepsilon$

$A \to S) \mid)S$

Exemplos: ((((((())))))), ()(), (()()), ()()(()(())).

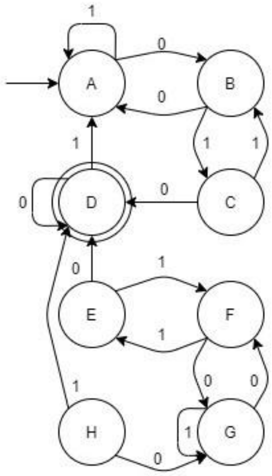
Exercícios do livro Hopcroft, Ullmann e Motwani

4.4.1 e 4.4.2) Minimizar os AFDs dados pelas tabelas de transições:

	0	1
→A	B	A
B	A	C
C	D	B
*D	D	A
E	D	F
F	G	E
G	F	G
H	G	D

1)

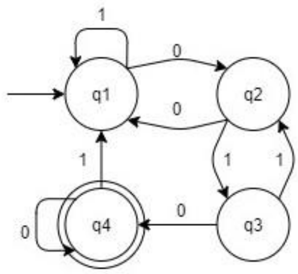
Os estados E, F, G e H são inúteis, pois a partir do estado inicial A, não é possível acessar esses estados. (pode ser visualizado pelo autômato ao lado)



Fazendo a tabela com os 4 estados não inúteis:

	q	0	1	π_0		0	1	π_1		0	1	π_2		0	1	π_3	
→	A	B	A	A		I	I	A	I	I	I	A	I		I	A	I
	B	A	C	B	I	I	I	B		I	II	B	II	I	III	B	II
	C	D	B	C		II	I	C	II	III	I	C	III	IV	II	C	III
☆	D	D	A	D	II	II	I	D	III	III	I	D	IV	IV	I	D	IV

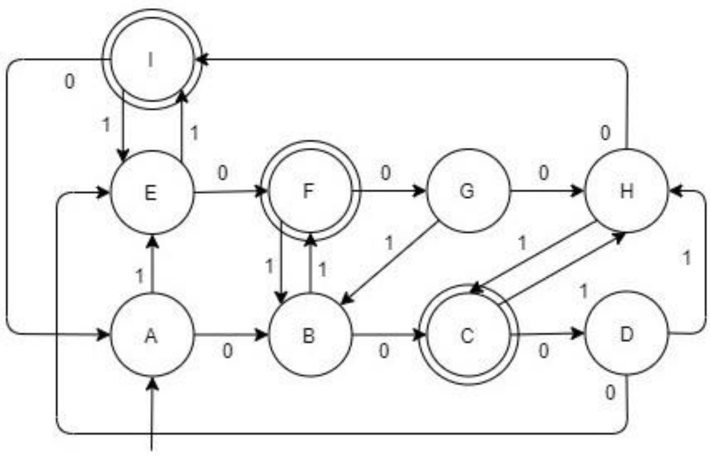
Com a tabela, geramos o autômato mínimo, que é o próprio autômato sem os estados inúteis.



	0	1
→A	B	E
B	C	F
*C	D	H
D	E	H
E	F	I
*F	G	B
G	H	B
H	I	C
*I	A	E

2)

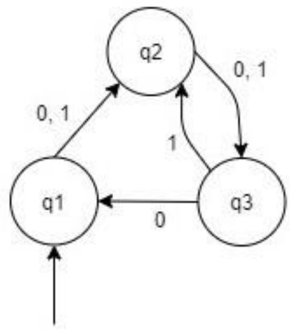
O autômato não apresenta estados inúteis.



Fazendo a tabela:

	q	0	1	π_0		0	1	π_1		0	1	π_2	
→	A	B	E	A		I	I	A		II	II	A	
	B	C	F	B		II	II	D	I	II	II	D	I
☆	C	D	H	D		I	I	G		II	II	G	
	D	E	H	E	I	II	II	B		III	III	B	
	E	F	I	G		I	I	E	II	III	III	E	II
☆	F	G	B	H		II	II	H		III	III	H	
	G	H	B	C		I	I	C		I	II	C	
	H	I	C	F	II	I	I	F	III	I	II	F	III
☆	I	A	E	I		I	I	I		I	II	I	

Com a tabela, geramos o autômato mínimo com somente 3 estados.



Exercícios do livro “Paulo Blauth Menezes. Linguagens Formais e Autômatos”

2.4 Desenvolva Expressões e Gramáticas Regulares que gerem as seguintes linguagens sobre $\Sigma = \{a, b\}$:

a) $\{ w \mid w$ tem no máximo um par de a como subpalavra e no máximo um par de b como subpalavra $\}$

ER: $((ab + ba)^* (aa + \varepsilon)(ab + ba)^* (bb + \varepsilon)(ab + ba)^*) +$

$((ab + ba)^* (bb + \varepsilon)(ab + ba)^* (aa + \varepsilon)(ab+ba)^*)$

GR: ({S, A}, {a, b}, P, S) onde P é:

$S \rightarrow XYXZX \mid XZXYX$

$X \rightarrow abX \mid baX \mid \varepsilon$

$Y \rightarrow aa \mid \varepsilon \quad Z \rightarrow bb \mid \varepsilon$

2.6 Descreva em palavras as linguagens geradas pelas seguintes Expressões Regulares:

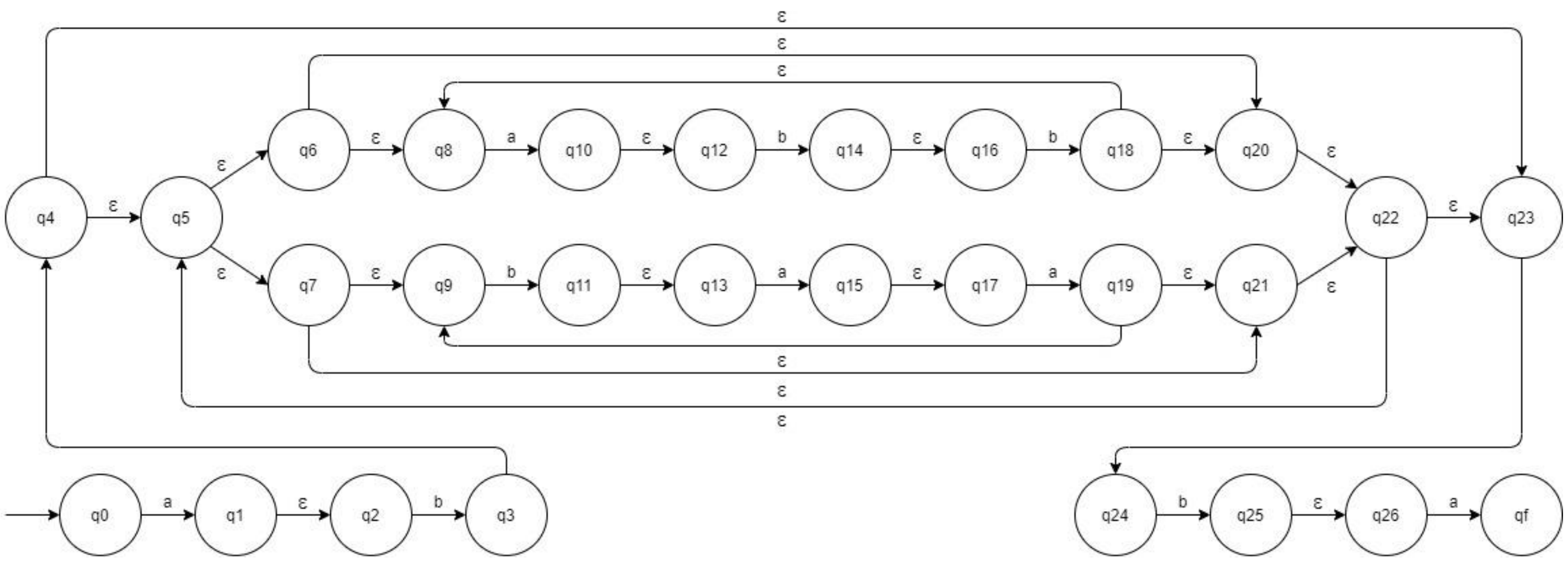
b) (b + ab)*(ε + a)

Exemplos: a, b, ab, ba, bb, aba, abb, bab, bba, bbb

L = {w | w não tenha a’s em sequências (2 ou mais ‘a’ juntos)}

2.7 Aplique o algoritmo de tradução de formalismo de Expressão Regular para Autômato Finito:

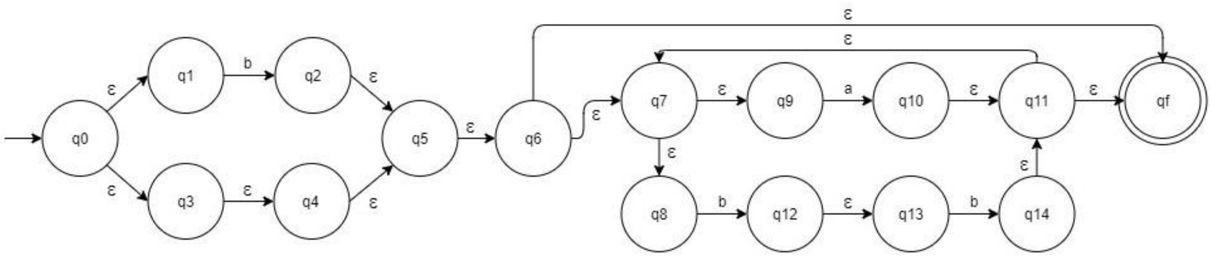
b) ab(abb* + baa*)*ba



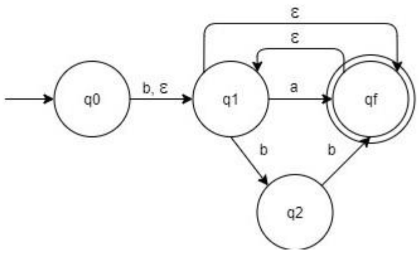
2.8 Aplique os algoritmos de tradução de formalismos apresentados e, a partir da Expressão Regular (b + ε)(a + bb)*, realize as diversas etapas até gerar a Gramática Regular correspondente (ER → AFε → AFN → AFD → GR).

ER : $(b + \varepsilon)(a + bb)^*$

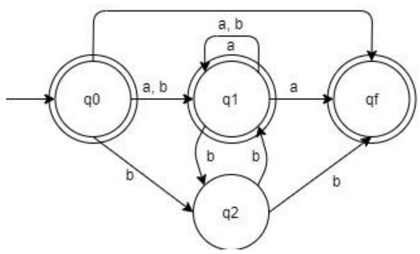
AFNe



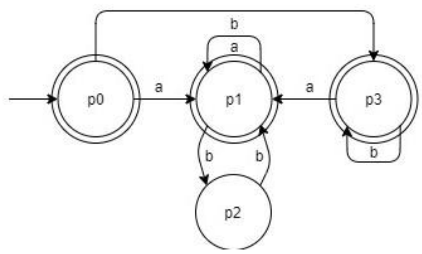
AFNe simplificado



AFND



AFD



GR: ({S, A}, {a, b}, P, S) onde P é:

$S \rightarrow b \mid bA \mid \varepsilon$

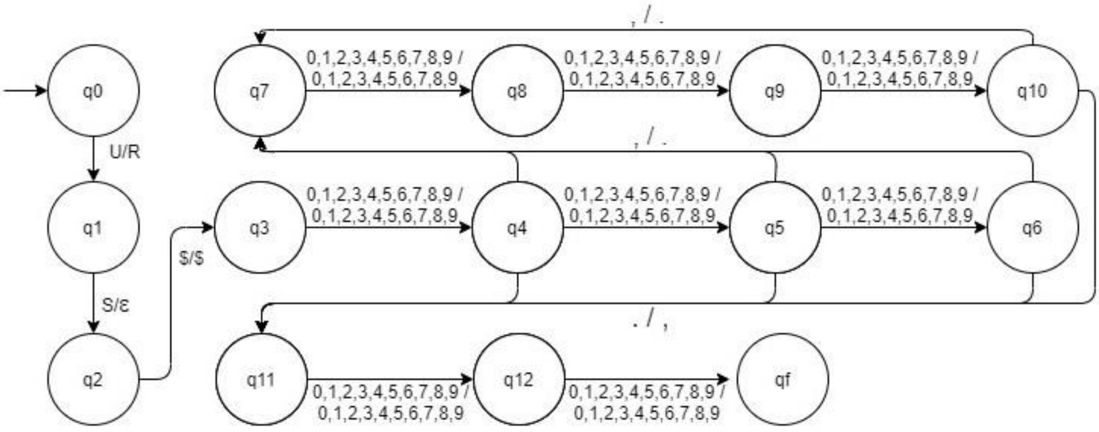
$A \rightarrow Aa \mid bbA \mid \varepsilon$

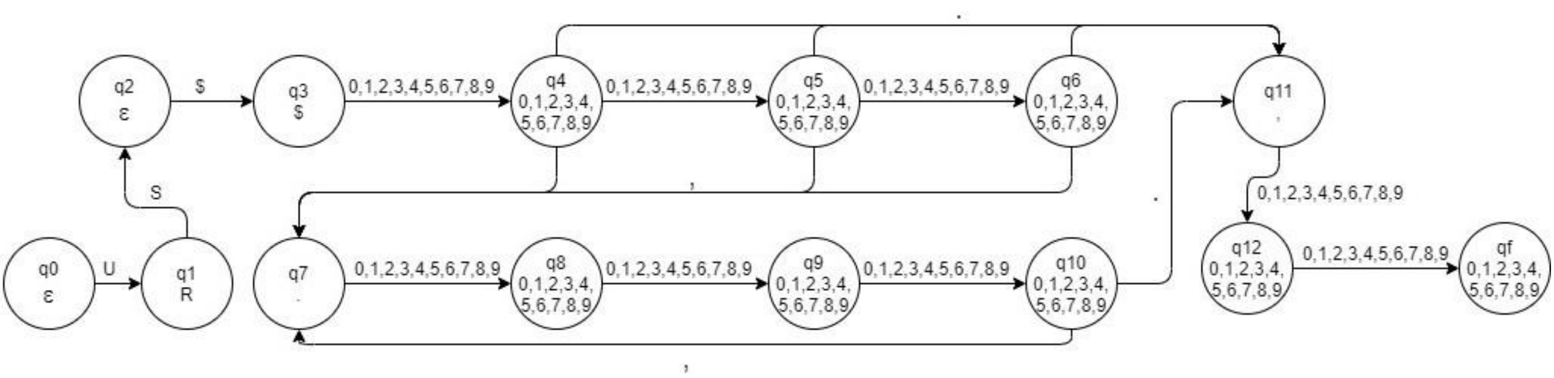
2.19 Desenvolva uma Máquina de Mealy e uma de Moore que realize a conversão da representação de valores monetários de dólares para reais. Por exemplo, dado o valor US$25,010.59 na fita de entrada, deve ser gravado o valor R$25.010,59 na fita de saída (atenção para o uso da vírgula e do ponto nos dois valores). Adicionalmente, o autômato deve verificar se a entrada é um valor monetário válido.

Utilizei 0,1,2,3... ao invés do d descrito na solução do analisador léxico, pois achei que facilitaria a visualização.

Sempre ao ler um número na fita de entrada o mesmo número será escrito na fita de saída. Utilizei isso para não precisar fazer 10 transições diferentes para cada conjunto de símbolo, semelhante ao descrito na solução.

O primeiro autômato é a maquina de Mealy e o segundo é o de Moore.





3.2 Desenvolva Gramáticas Livres do Contexto que gerem as seguintes linguagens:

d) $L_4 = \{ w \mid w$ é palíndromo em $\{a, b\}^* \}$, onde palíndromo significa que $w = w^r$

f) $L_6 = \{ a^i b^j c^k \mid i = j$ ou $j = k$ e $i, j, k \geq 0 \}$

G4 = ({S, A, B}, {a, b}, P, S) onde P é

$S \rightarrow aA \mid bB \mid a \mid b \mid \varepsilon$

$A \rightarrow Sa$

$B \rightarrow Sb$

Exemplos: aa, abba, baaab, aaaaa, bababab.

G6 = ({S, A, B, C, D, E}, {a, b, c}, P, S) onde P é

$S \rightarrow B$

$B \rightarrow aDbC \mid AbEc$

$A \rightarrow aA \mid \varepsilon$

$C \rightarrow Cc \mid \varepsilon$

$D \rightarrow aDb \mid \varepsilon$

$E \rightarrow bEc \mid \varepsilon$

Exemplos: aaabbbc, abbbccc, aabbcc, abccc.

3.11 a) Optei por não encaminhar esse exercício

3.16 As linguagens geradas pelas gramáticas cujas produções estão representadas abaixo são vazias, finitas ou infinitas?

a)

$S \rightarrow AB \mid CA$
$A \rightarrow a$
$B \rightarrow BC$
$C \rightarrow AB \mid \varepsilon$

b)

$S \rightarrow aS \mid aSbS \mid X$
$X \rightarrow SS$

A linguagem da gramatica 1 é finita. Se o autômato encontrar o estado B, ela fica infinita, mas a partir de S começando com CA o C pode ser cadeia vazia, resultando em um único a, sem encontrar o loop infinito de B.

Já a segunda gramatica é infinita. S sempre retorna pra S ou X, mas X sempre volta para S, sem nenhuma saída, sendo infinita.

OBS:

Não consegui encontrar o exercício 2.24 do livro do Hopcroft e Ullman também não consegui fazer o exercício 3.11 (a) do livro do Paulo Blauth.

Caso alguma imagem não tenha ficado boa para visualização, estou deixando um link do drive para o arquivo que fiz os autômatos, usando o site draw.io

https://drive.google.com/drive/folders/1pXaQAgn6JwxiZ07Bn9f9i4AtrkbshvPi?usp=sharing

Os arquivos 'AFs' e 'Afs2' contém todos os autômatos juntos.