Linguagens Formais e Autômatos

Arthur do Prado Labaki – 11821BCC017

Segunda Prova

1ª Questão

Seja a linguagem L = { w ∈ {0,1}* | o número de 1s é **maior** que o número de 0s e existe pelo menos uma ocorrência de 0}.

Ex 01101, 1110, 010111, 10101, etc.

Prove que essa linguagem **não** é regular usando o **Lema do Bombeamento para linguagens regulares**, utilizando o roteiro abaixo.

1- $L = \{w \in (0,1)^* \mid |1| > |0| \text{ e } |0| \geq 1\}$

w pode ser definida como uvz, em que $|w| \geq n$
$|uv| \leq n$
$|v| \geq 1$
Para todo $i \geq 0$, $uv^iz \in L$

$1^{n+1} \; 0^n$

$\overbrace{111\ldots}^{|uv| \text{ máximo}}$ 100...

$\underbrace{111\ldots 1}_{n+1}\underbrace{00\ldots}_{n}$

Supondo que L é regular, o lema do bombeamento é valido.

Escolhendo a palavra $1^{n+1}0^{n}$ pois ela pertence a linguagem e irá facilitar nas quebras de w.

Sendo w = uvz, onde |uv| <= n e |v| >= 1 em todo i >= 0, $uv^{i}z$

Como n+1 > n |uv| contem somente números 1's

Bombeando em w, temos w' = uz, em que i = 0.

Logo w' será $1^{n+1-r}0^{n}$ não pertencendo a palavra escolhida, pois n+1-r é menor que n, pois r tem seu tamanho mínimo 1 (n+1-r < n).'

Com essa prova, concluímos, por absurdo, que a linguagem não é regular.

2ª Questão

a) Simplifique a gramática a seguir, eliminando as produções vazias (apresentar todos os passos do algoritmo).

G=({S,X,Y,Z},{a,b},P,S) sendo

P:{S→ XY|Yb|aX, X→ ξ|bb|aSa, Y→ Z|ZX|Zba, Z→ ξ|b|YY|aZ}

b) Na gramática resultante do item (a), remover os símbolos inúteis (apresentar todos os passos do algoritmo).

Optei por não fazer essa questão.

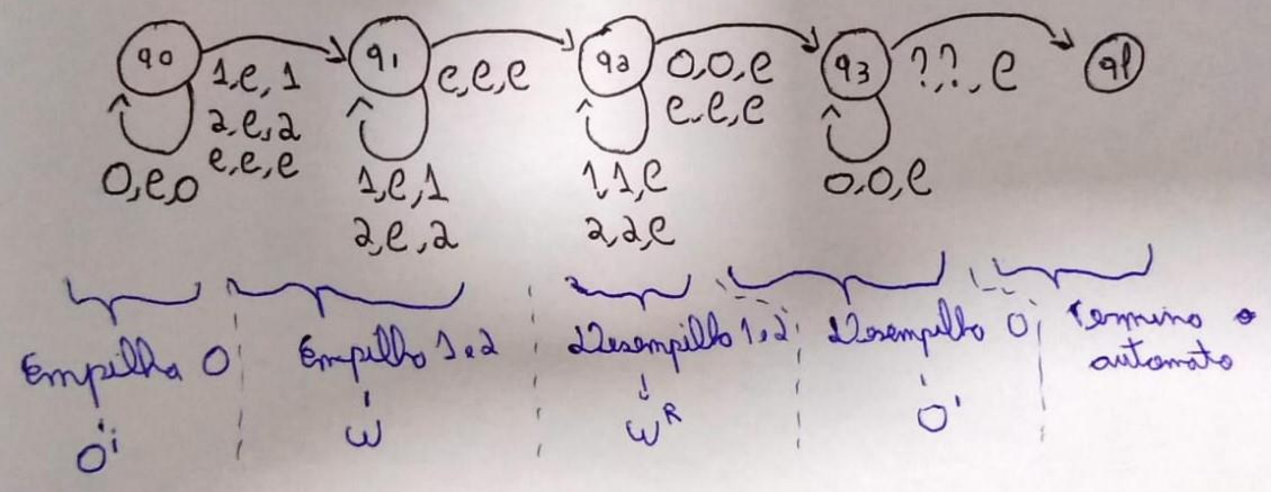
3ª Questão

Escreva um Autômato de Pilha que conheça a linguagem a seguir.

Ao final diga se o AP construído é determinístico ou não determinístico.

L = { $w_1 \in \{0,1,2\}^*$ | $w_1 = 0^i w w^R 0^i$, sendo $i \geq 0$, $w \in \{1,2\}^*$ e w^R representa o reverso de w}.
Ex: 01222210, 002200, 11211211, 0000, etc.

3 - $0^i w w^R 0^i$



É uma automato de Pilha não deterministico, pois existe estados (como q0) que realizam mais de uma transição.

4ª Questão

Seja a gramática a seguir para gerar expressões lógicas:

$G=(\{S\}, \{\wedge, \vee, \rightarrow, (,), a, b, c\}, P, S)$ e $P = \{S \rightarrow S\wedge S \mid S\vee S \mid S\rightarrow S \mid (S) \mid a \mid b \mid c)$

a) Remova a ambiguidade considerando as seguintes precedências para os operadores lógicos: $\{\rightarrow\} > \{\wedge, \vee\}$

b) Explique quais são as ambiguidades na G original e como você as removeu no item a.

c) Mostre todas as árvores de derivação possíveis nas duas gramáticas (ambígua e não ambígua) para a expressão: $a \rightarrow (b \vee a \wedge c)$

4- a) $G_2=(\{S,T,F,I\},\{\wedge,\vee,\rightarrow,(,),a,b,c\},P_2,S)$

$P_2=\{S\rightarrow T \mid S\wedge T \mid S\vee T,$

$T\rightarrow F \mid T\rightarrow F,$

$F\rightarrow I \mid (S),$

$I\rightarrow a \mid b \mid c\}$

b) A ambiguidade se da no fato que os operadores não tem uma ordem de procedência, o que leva a gerar mais de uma arvore de derivação.

Eu removi essa ambiguidade colocando procedência neles, em que se consistiu em criar novos estados, separando-os.

S permanece os primeiros operadores

T o operador '->'

F os operadores '(' e ')'

I o alfabeto (a, b, c)

c) $a \to (b \vee a \wedge c)$