Questão 1. Dê a gramática para as seguintes linguagens:

a. $\{ a^n b^n \mid n \in N \}$ → $S \to aSb \mid \lambda$

S+0=Sbs

b. $\{ a^n b^k c^m \mid k = n+m \}$ → $S \to BC$, $B \to aBb \mid \lambda$, $C \to bCc \mid \lambda$

c. $\{ a^n b^k c^m \mid k = 2n+m \}$ ✓

d. $\{ a^m b^n c^i \mid m > n + i \}$ ✓

e. $\{ w \in \{ a, b \}^* \mid w$ não contenha o substring aba $\}$ ✓

f. $\{ w \mid w$ contenha um número igual de a´s e b´s $\}$ ✓

g. $\{ ww^R \mid w \in \{ a, b \}^* \}$ ✓

Questão 2. Para cada uma das gramáticas a seguir, descreva a linguagem gerada pela gramática:

aaaabb

a. $S \to aaSB \mid \lambda$
$B \to bB \mid \lambda$
$L = (aa^+ b^* \mid a^* \mid \lambda)$ ou $L = \{ w \mid w = \lambda, (aa)^n b^m / n > 0, m \geq 0 \}$

aaccbb

b. $S \to aSbb \mid A$
$A \to cA \mid c$
$L = (a^+ c^+ (bb)^+ \mid c)$ ou $L = \{ w \mid w = c, a^m c^n b^{2m} \mid n > 0, m \geq 0 \}$

c. $S \to aS \mid bS \mid A$
$A \to cA \mid c \mid \lambda$
$L = \{ w \mid w \in \{a, b, c\}^*$, se 'c', não pode + colocar a ou b$\}$
ou : $[(a \cup b)^* c^* \cup (a \cup b)^* cc^+]$?

d. $S \to \boxed{ab}S\boxed{dc} \mid A$
$A \to \boxed{cd}A\boxed{ba} \mid \lambda$
$L = \{ (ab)^n (cd)^m (ba)^m (dc)^n \mid n, m \geq 0 \}$?

e. $S \to aA \mid \lambda$
$A \to bS$
$L = \{ w \mid w = \lambda, (ab)^+ \}$ ou $L = \{ w \mid w \in \{ab\}^* \}$ ou $L = \{ (ab)^n, n \geq 0 \}$

ababab (sempre q tem um 'a', tem um 'b').

Questão 3. Para cada uma das gramáticas a seguir, obtenha uma gramática essencialmente não contrátil. → só o símbolo inicial tem produção λ (e os outros λ foram tratados)

a. $S \to aS \mid bS \mid B$
$B \to bb \mid C \mid \lambda$
$C \to cC \mid \lambda$

b. $S \to ABC \mid \lambda$
$A \to aA \mid \lambda$
$B \to bB \mid \lambda$
$C \to cC \mid \lambda$

c. $S \to BSA \mid A$
$A \to aA \mid \lambda$
$B \to Bba \mid \lambda$

$b = 2a + c$

$\boxed{2b = a} + c$

c. $\{ a^n b^k c^m \mid k = 2n+m \}$

$S \to AB$
$A \to aAbb \mid \lambda$
$B \to bBc \mid \lambda$

d. $\{ a^m b^n c^i \mid m > n + i \}$

$a > b + c$

a tem que
m ao menos = 1

$a^i \; a^m \; b^i \; c^m$
LLC

$\cancel{S \to aABC}$
$\cancel{A \to aA \mid \lambda}$
$\cancel{B \to aCc \mid C \mid \lambda}$
$\cancel{C \to aCb \mid \lambda}$

a a c
a
a λ

$S \to a \mid aS \mid aB$
se quiser repetir o a
se quiser sair do loop de a
$B \to aBc \mid A$
$A \to aAb \mid \lambda$

BABAB ✗

e. $\{ w \in \{a, b\}^* \mid$ w não contenha o substring aba $\}$

~~S → AB | BA | BAB | ACA~~
~~A → Aa | λ~~
~~B → Bb | λ~~
~~C → bbB~~

corrigindo

$S' \to A \mid b \mid S$
$S \to \cancel{ABAS}$
$A \to \cancel{Aa \mid \lambda}$
$B \to \cancel{Bb \mid bb}$

ABAS
bb ABA

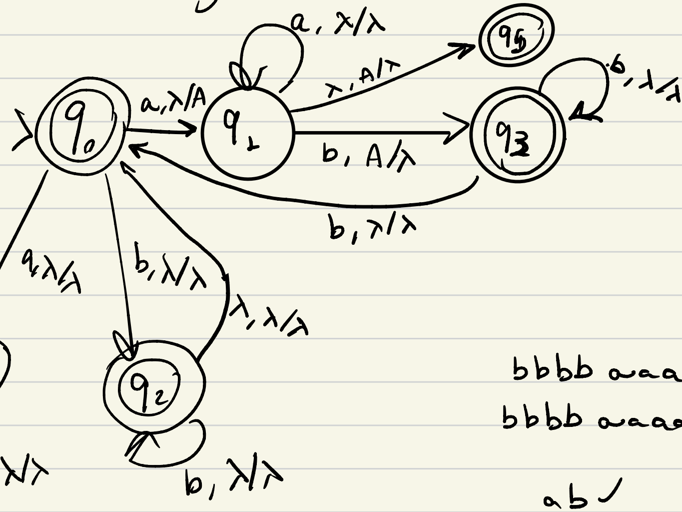
aaabb
bbaaa
bbbaacbb
aaab
abab
bacaab
aaaab
abbbbbaaa
ab
abbabba
aab
a
b
abbbaa
ba
bbba
BAB
abbbbbaaab
abbabba

S
a a b
a b b b
a a a a
b b b
b a a b
b a b
a a b
ba
✓

aba
aaabaa
abaaa

se começar c/ a, e tiver b n̄ pode ter outro a

Fazendo o autômato AP



A

bbbb aaaabb

bbbb aaaabbbaaabb

ab ✓

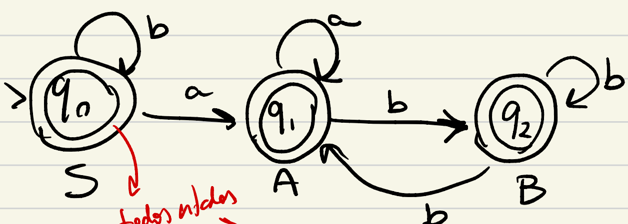
ba ✓

abbabba

A | _ | | a ~~A~~ | | _

q_1

Fazendo AF p/ descobrir a gramática (ma é LR):

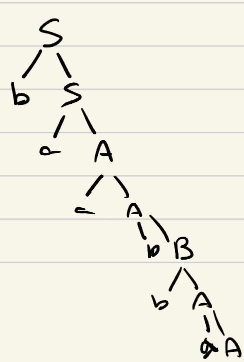


todos estados podem ser λ

$S \to bS \mid aA \mid \lambda$

$A \to aA \mid bB \mid \lambda$

$B \to bB \mid bA \mid \lambda$



aaabbb

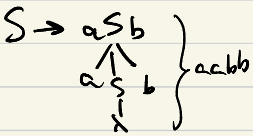
f. { w | w contenha um número igual de a´s e b´s }

$S \to aSb \mid bSa \mid A \mid \lambda$

$A \to abA \mid baA \mid \lambda$

aabb →
bbaa
baab
aabb
abab
baba

$S \to aSb \mid bSa \mid baS \mid abS \mid \lambda$



$S \to abS \to abS \to \lambda = abab$

$S \to baabba$

abaabb

ab

baabba

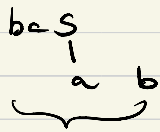
$baS \to abS \to baS$ ST
ou
$bSa \to aSb \to abS \to \lambda$

ambiguidade

SIMPLIFICANDO

baabb a

$S \to baS \mid abS \mid aS_1b \mid \lambda$



baab
abba
$abS \to baS \to \lambda$

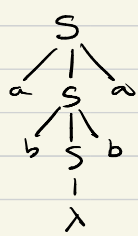
$S \to baS \mid abS \mid aS_1b \mid \lambda$
$S_1 \to aS_1b \mid \lambda$

⇒ PRECISA SER ASSIM??

g. { ww^R | w ∈ {a, b}* }

abba
bbaa
abaaba

$S \to aSa \mid bSb \mid \lambda$



3- a) $S \to aS \mid bS \mid B$
$B \to bb \mid C \mid \lambda$
$C \to cC \mid \lambda$

$\Downarrow$

$S' \to S \mid \lambda$
$S \to aS \mid bS \mid B$
$B \to bb \mid C \mid \cancel{\lambda}\, b$
$C \to cC \mid c$

trocando pelo c zinho, é o mesmo impacto

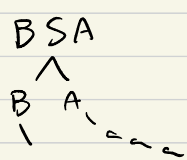
b) $S \to ABC \mid \lambda$
$A \to aA \mid \lambda$
$B \to bB \mid \lambda$
$C \to cC \mid \lambda$

$\Downarrow$

$S' \to S \mid \lambda$
$S \to ABC \mid AB \mid AC \mid BC \mid A \mid B \mid C$
$A \to aA \mid a$
$B \to bB \mid b$
$C \to cC \mid c$

meio que anula a existência dele antes

$S \to BSA \mid A$
podem virar T $\begin{cases} A \to aA \mid \lambda \\ B \to Bba \mid \lambda \end{cases}$



$\Downarrow$

$S' \to S \mid \lambda$
$S \to BSA \mid A \mid B \mid BS \mid SA \mid BA$
$A \to aA \mid a$
$B \to Bba \mid ba$

um não terminal produzindo um outro não-terminal

Questão 4. Para cada uma das gramáticas a seguir, obtenha uma gramática equivalente sem regras de cadeia. → eliminação de redundância (remover as produções A → B)

a. S → AS | ~~A~~ (regra de cadeia; aA | bB | cC | b)
A → aA | bB | ~~C~~ cC | b
B → bB | b
C → cC | ~~B~~ bB | b

⇒

S → AS | ~~A~~ aA | bB | cC | bB | b
A → aA | bB | ~~C~~ cC | bB | b
B → bB | b
C → cC | ~~B~~ bB | b

F(S) = {S, A, C, B}
F(A) = {A, C, B}
F(C) = {C, B}

melhor começar c/ o menor fecho (depende de menos variáveis)

1 - cria S' q chama S ou S ⇒
2 - eliminar produções λ
3 - eliminar regra de cadeia
4 - eliminar inúteis (inacessíveis)
5 - binarização

b. S → A | B | C
A → aa | B
B → bb | C
C → cc | A

⇒

S → ~~A~~ | ~~B~~ | ~~C~~ aa | bb | cc
A → aa | ~~B~~ bb | cc
B → bb | ~~C~~ cc | aa
C → cc | ~~A~~ aa | ~~B~~ bb | ~~C~~

Questão 5. Para cada uma das gramáticas a seguir, obtenha uma gramática equivalente na FNC.

f. S → aAbB | ABC | a
A → aA | a
B → bBcC | b
C → abc

⇒

S' → aAbB | ABC | a
~~S → aAbB | ABC | a~~ inalcançável
A → aA | a
B → bBcC | b
C → abc

onde o comprimento é > 1.

⇒

S' → $\underbrace{X_a \overbrace{A \underbrace{X_b B}_{T2}}^{T1}}$ | A$\underbrace{BC}_{T3}$ | a
A → X_aA | a
B → X_bB$\underbrace{X_c C}_{T4}$ | b
C → $X_a\underbrace{X_b X_c}_{T5}$
X_a → a
X_b → b
X_c → c
T1 → AT2
T2 → X_bB
T3 → BC
T4 → X_cC
T5 → $X_b X_c$

g. S → A | ABa | AbA
A → Aa | λ
B → Bb | BC
C → CB | CA | bB

h. S → ABC
A → a
B → b | bb
C → BaB | c

i. S → ADE | ABa | AbA
A → Aa | λ
B → Bb | BC
C → CB | CA | bB
D → EdD | E
E → bcdE | D

Questão 6. Prove usando o pumping lemma que as seguintes linguagens não são livres de contexto:

a. { $a^n b^n a^n b^n$ | n > 0 }

b. { $0^n 1^{2n} 2^n$ | n > 0 }

c. { $ww^R w$ | w ∈ { a, b }* }

Questão 7. Construa autômatos de pilha que reconheçam as seguintes linguagens:

a. { $w0w^R$ | w ∈ { a, b }* }

b. { $a^n b^n a^m$ | n, m ≥ 0 }

c. { w | w ∈ { a, b, c }* e o primeiro c seja precedido por aaa }

S pode ser anulável ✓

g) $S \to A \mid ABa \mid AbA$
$A \to Aa \mid \lambda$
$B \to Bb \mid BC$
$C \to CB \mid CA \mid bB$

$\Rightarrow$

$S' \to S \mid \lambda$
$S \to A \mid ABa \mid AbA \mid Ba \mid b \mid bA \mid Ab$
$A \to Aa \mid \cancel{\lambda}\, a$ → a é anulável, então
$B \to Bb \mid BC$
$C \to CB \mid CA \mid bB$

(regra de cadeia)

$S' \to \cancel{S} \mid \lambda \mid \cancel{ABa} \mid AbA \mid \cancel{ABa} \mid bA \mid Ab \mid b$
~~$S \to A \mid ABa \mid AbA \mid Aa \mid a \mid Ba \mid b \mid bA \mid Ab$~~
$A \to Aa \mid \cancel{\lambda}\, a$
~~$B \to Bb \mid BC$~~
~~$C \to CB \mid CA \mid bB$~~

não produzem terminais

não é atingível

$S' \to A\underbrace{\cancel{b}A}_{T_1} \mid \cancel{b}A \mid A\cancel{b} \mid b \mid \lambda$
$A \to A\cancel{a} \mid \cancel{\lambda}\, a$
$X_a \to a$
$X_b \to b$
$T_1 = X_bA$

$S' \to AT_1 \mid X_bA \mid AX_b \mid b \mid \lambda$
$A \to AX_a \mid a$
$X_a \to a$
$X_b \to b$
$T_1 \to X_bA$

h. $S \to ABC$
$A \to a$
$B \to b \mid bb$
$C \to BaB \mid c$

1°

h) $S \to ABC$
$A \to a$
$B \to b \mid bb$
$C \to BaB \mid c$

$\Rightarrow$

$S' \to S$
$S \to ABC$
$A \to a$
$B \to b \mid bb$
$C \to BaB \mid c$

2° → não tem λ

3° eliminar regras de cadeia

$S' \to \cancel{S} \mid ABC$
~~$S \to ABC$~~
$A \to a$
$B \to b \mid bb$
$C \to BaB \mid c$

$F(S') \to \{S', S\}$

S = inútil (inalcançável)

todos produzem terminais (direta ou indiretamente)

4° isolar term.

$S' \to \cancel{S} \mid ABC$
~~$S \to ABC$~~
$A \to a$
$B \to b \mid \cancel{bb}\ X_bX_b$
$C \to BAB \mid c$
$X_b \to b$
$X_c \to c$

5° quebrar cadeias longas

$S' \to \cancel{S} \mid A\cancel{BC}T_1$
~~$S \to A\cancel{BC}T_1$~~
$A \to a$
$B \to b \mid X_bX_b$
$C \to B\underbrace{AB}_{T_2} \mid c$
$X_b \to b$
$X_c \to c$
$T_1 \to BC$
$T_2 \to AB$

Final

$S' \to \cancel{S} \mid AT_1$
~~$S \to AT_1$~~
$A \to a$
$B \to b \mid X_bX_b$
$C \to BT_2 \mid c$
$X_b \to b$
$X_c \to c$
$T_1 \to BC$
$T_2 \to AB$

i. $S \to ADE \mid ABa \mid AbA$
$A \to Aa \mid \lambda$
$B \to Bb \mid BC$
$C \to CB \mid CA \mid bB$
$D \to EdD \mid E$
$E \to bcdE \mid D$

1° e 2° passo:

$S \to ADE \mid ABa \mid AbA$
$A \to Aa \mid \lambda$
$B \to Bb \mid BC$
$C \to CB \mid CA \mid bB$
$D \to EdD \mid E$
$E \to bcdE \mid D$

$\Rightarrow$

$S' \to S$
$S \to ADE \mid ABa \mid AbA \mid DE \mid Ba \mid Ab \mid bA \mid b$
$A \to Aa \mid \cancel{\lambda}\ a$
$B \to Bb \mid BC$
$C \to CB \mid CA \mid bB \mid \cancel{\lambda}$
$D \to EdD \mid E$
$E \to bcdE \mid D$

$\Downarrow$ 3° (λ cadeia)

$S' \to \cancel{S} \ldots \mid AbA \ldots \mid Ab \mid bA \mid b$
$\cancel{S \to ADE \mid ABa \mid AbA \mid DE \mid Ba \mid Ab \mid bA \mid b}$
$A \to Aa \mid \cancel{\lambda}\ a$
$\cancel{B \to Bb \mid BC}$
$\cancel{C \to CB \mid CA \mid bB \mid \lambda}$
$\cancel{D \to EdD \mid E \mid bcdE}$
$\cancel{E \to bcdE \mid D \mid EdD}$

4° passo:

1° produção terminais (direto/ind.): $\{S', S', A\}$

2° não alcançáveis

alcançáveis a partir de S' $= \{S', A\}$

$\Downarrow$

remove o S

$F(S) = \{S', S\}$
$F(D) = \{D, E\}$
$F(E) = \{E, D\}$

5° passo

$S' \to A\cancel{X_b}A \mid A\cancel{b} \mid \cancel{b}A \mid b$
$A \to A\cancel{X_a} \mid a$
$X_a \to a$
$X_b \to b$

$\Downarrow$ 6° passo

$$\boxed{\begin{array}{l} S' \to AT_1 \mid AX_b \mid X_bA \mid b \\ A \to AX_a \mid a \\ X_a \to a \\ X_b \to b \\ T_1 \to X_bA \end{array}}$$

Questão 6. Prove usando o pumping lemma que as seguintes linguagens não são livres de contexto:

a. $\{ a^n b^n a^n b^n \mid n > 0 \}$

$L = \{a^n b^n a^n b^n \mid n > 0\}$

$\to \ p \mid a^n \ q \ b^n \mid v \ a^n \ t \ b^n \mid y$

não é possível dividir as substrings de $L(G)$ de tal forma que, bombeando q e t, o nº de a's e b's sejam iguais e permaneçam na ordem de linguagem.

$a \mid a^{2^k/4 - 4} \mid ab \mid b^{2^k/4 - 4} \mid ba \mid a^{2^k/4 - 4} \mid ab \mid b^{2^k/4 - 4} \mid b$

$p_2 \ q_1 \ p_1 \ q_2 \ v \ t_1 \ y \ t_2 \ y_2$

deveríamos ter 2 q's e 2 t's p/ bombear igualmente

$p \mid a^n \ q \ b^{n-1} \mid ab \ v \mid a^{n-1} \ t \ b^n \mid y$

$a b a b a b \boxed{ab} a b a b$

não obriga a ter +1 ab

$p \mid a^n b^n \mid a^n b^n \Rightarrow$ ab ab ab ab ab $\mid$ ab
$\quad\; qvt \qquad\quad y$

$n = 0$

$|ab^0 \neq 0$

Se $n = 0$

$a^0 b^0 ab \; a^0 b^0$

$ab \neq 0$

b. $L = \{ 0^n 1^{2n} 2^n \mid n > 0 \}$

$0^n 1^{2n} 2^n$

$n + n = 2n$
$0 \quad 2 \quad\quad 1$

Se $L(G)$ é uma LLC, gerada por uma GLC-FNC com K variáveis, um nº K deve existir tal que todos os strings w em L com $|w| > 2^{K-1}$ possam ser decompostas satisfazendo a LLC.

Tome $w = 0^{2^K} 1^{2 \cdot 2^K} 2^{2^K}$. Como $|w| > 2^{K-1}$ ($|w| = 2 \cdot 2^K = 2^{K+1}$), pelo lema do bomb:

1) $w = pqvty$
2) $|q| + |t| > 0$
3) $|qvt| \leq 2^{K-1}$
4) $w_i = pq^i v t^i y \in L(G), \forall i \geq 0$

1º CASO

$$p \mid \underbrace{0^{2^K}}_{qvt} \mid \underbrace{1^{2 \cdot 2^K} \quad 2^{2^K}}_{y}$$

bombeando: o nº de 0's +, 1's e 2's =. $w_i \notin L(G)$

análogo p/ 1's e 2's

perde a proporção

2º CASO

$$p \mid \underbrace{0^{2^{K-1}}}_{q} \mid \underbrace{01}_{v} \mid \underbrace{1^{2 \cdot 2^{K-1}}}_{t} \mid \underbrace{2^{2^K}}_{y}$$

bombeando: nº de 0's e 1's +, mas de 2's =. $w_i \notin L(G)$

análogo p/ 1's e 2's, impossível p/ 0's e 2's.

3º CASO

$$p \mid \underbrace{0^{2^K}}_{q} \quad \underbrace{1^{2 \cdot 2^K}}_{v \;\; t} \quad 2^{2^K} \mid y$$

bombeando os 3 juntos: impossível! pois $|qvt| > 2^{K-1}$.

PROVA POR ABSURDO

2.1º CASO

$$p \mid \underbrace{0^{2^K}}_{q} \quad \underbrace{1^{2 \cdot 2^K}}_{v \;\; t} \mid \underbrace{2^{2^K}}_{y}$$

bombeando 0's e 1's simultaneamente = a ordem é desfeita, pois o bombeamento resultará em 0101202...
$w_i \notin L(G)$.

análogo p/ 1's e 2's
impossível p/ 0's e 2's.

$\underbrace{ab}_{w}\ \underbrace{ba}_{w^R}\ \underbrace{ab}_{w}$

c. $\{ww^Rw \mid w \in \{a, b\}^*\}$

~~$w^{2^K} w^{R^{2K}} w^{2K}$~~

Se $L(G)$ é uma LLC gerada por uma GLC-FNC com K variáveis, existe um comprimento de bombeamento p (onde $p = 2^{K-1}$), em que todos os strings w em $L(G)$ com $|w| > 2^{K-1}$ devem poder ser decompostos e permanecer satisfazendo a $L(G)$.

(escolhendo uma string de comprimento suficientemente grande)

Dado $a^{2^{K-1}} b^{2^{K-1}} b^{2^{K-1}} a^{2^{K-1}} a^{2^{K-1}} b^{2^{K-1}}$, Como $|w| > 2^{K-1}$ $(|w| = 2 \cdot 2^K = 2^{K+1})$, pelo L.B:

I) $w = pqvty$

II) $|q| + |t| > 0$

III) $|qvt| \leq (2^{K-1}) \cdot \cancel{c}$

IV) $w_i = pq^ivt^iy \in L(G), \forall i \geq 0$

isso não muda pq K é o nº de níveis da AD, o que independe da qntd de a's e b's!

$\left.\begin{matrix} w_1, w_2 = a^{2^K} b^{2^K} \\ w_{1R} = b^{2^K} a^{2^K} \end{matrix}\right\} w$

$w = w_1 w_{1R} w_2 = \underbrace{a^{2^K} b^{2^K}}\ \underbrace{b^{2^K} a^{2^K}}\ \underbrace{a^{2^K} b^{2^K}}$

3 subseções críticas (que precisam ser bombeadas na mesma qntd) p/ $w_i \in L(G)$

1º CASO: q e t em só uma variável

$\underset{p}{}\,|\underset{qvt}{a^{2^{K-1}}}|\, b^{2^{K-1}} \underbrace{b^{2^{K-1}} a^{2^{K-1}} a^{2^{K-1}} b^{2^{K-1}}}_{y}$

bombeando, a qntd de a's de w_1 +, a do restante não. $|w_1| \neq |w_{1R}|$ e $w_1 \neq w_2$

- análogo p/ todos as outras variáveis.

4º CASO:

$\underset{p}{}\,|\underset{q}{a^{2^{K-1}}}|\underset{v}{b^{2^{K-1}}}|\underset{t}{b^{2^{K-1}}}|\, \underbrace{a^{2^{K-1}} a^{2^{K-1}} b^{2^{K-1}}}_{y}$ = bombeando *

2º CASO: q e t em w_R

$\underset{p}{a^{2^{K-1}}}\,|\underset{q\,v\,t}{b^{2^{K-1}} b^{2^{K-1}}}|\, \underbrace{a^{2^{K-1}} a^{2^{K-1}} b^{2^{K-1}}}_{y}$

bombeando, a string w_R fica desbalanceada

3º CASO: q e t em w_R

$\underset{p}{a^{2^{K-1}} b^{2^{K-1}}}\,|\underset{q\,v\,t}{b^{2^{K-1}} a^{2^{K-1}}}|\, \underbrace{a^{2^{K-1}} b^{2^{K-1}}}_{y}$

bombeando, $|w_1| \neq |w_R|$

análogos

5º CASO:

$\underset{p}{a^{2^{K-1}} b^{2^{K-1}} b^{2^{K-1}} a^{2^{K-1}}}\,|\underset{q\,v\,t}{a^{2^{K}} b^{2^K}}|\,\underset{y}{}$

bombeando, $w_1 \neq w_2$

6º CASO:

$\underset{p}{a^{2^{K-1}} b^{2^{K-1}}}\,|\underset{q\,v\,t}{b^{2^{K-1}} a^{2^{K-1}} a^{2^{K}} b^{2^K}}|\,\underset{y}{}$

* a condição de $L(G)$ é que $w_1 = w_2$, e $|w_1| = |w_R|$ mas bombeando q e t, $w_1 \gtrless w_2$, então $w_i \notin L(G)$ ou $|w_1| \neq |w_R|$

* analogamente p/ qualquer combinação de $2 < |v| < 6$ $w_1 \neq w_2$ e $|w_1| \neq |w_R|$

$w_2 > w_1$ e $|w_R| \neq |w_1|$

PROVA POR ABSURDO!!

7° CASO:

$$a^{2^{k-1}}\,b^{2^{k-1}}\,b^{2^{k-1}}\,a^{2^{k-1}}\,a^{2^{k-1}}\,b^{2^{k}}$$

q v t   y

↳ bombeando: IMPOSSÍVEL! pois $|qvt| > (2^{k-1}) \cdot 6$

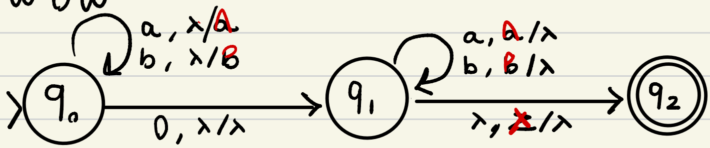
Questão 7. Construa autômatos de pilha que reconheçam as seguintes linguagens:

a. $\{w0w^R \mid w \in \{a, b\}^*\}$

aabbaa

$w0w^R$ ← reverso

(palíndromo c/ 0 no meio)



aba0aba

a b a 0 a

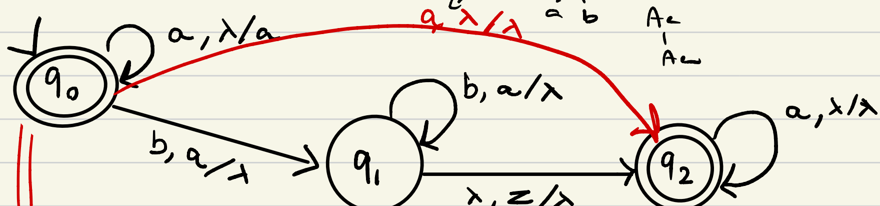
Z → MerceFundo

PRECISA DO MF AQUI?

ele só vai aceitar se tiver vazia a pilha!

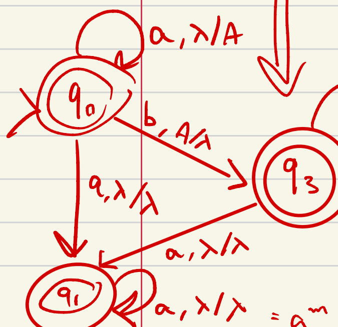
b. $\{a^nb^na^m \mid n, m \geq 0\}$

pode terminar aq pq a agnt quer m=2 e n=0 p/ ex, só temos 2 as e aceitamos



S → a S b | A a

$a^nb^na^m$



aaa bbb aa

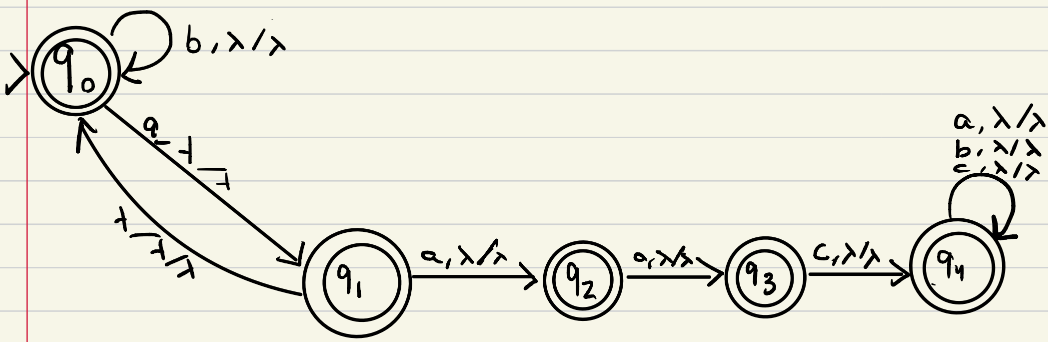
a a a b b b λ a

bbbb a

c. { w | w ∈ { a, b, c }* e o primeiro c seja precedido por aaa }

antes!

aaaa ?



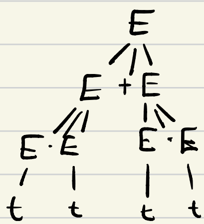
aaacbcbc

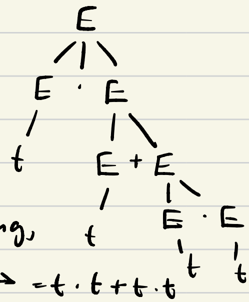
abcb

ababaaaac

Questão de Ambiguidade de Gramática:

1) $G_1: E \to E + E \mid E \cdot E \mid t$



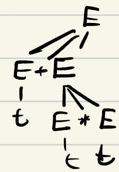


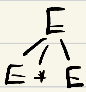
mesma string, ADs $\neq$.

$= t \cdot t + t \cdot t \longleftrightarrow = t \cdot t + t \cdot t$

outra string $\Rightarrow$ $w = t + t \cdot t$

(violando a precedência)





$E \to E + E \to t + E \to t + E * E \to t + t * E \to t + t * t$

$E \to E * E \to E + E * E \to t + E * E \to t + t * E \to t + t * t$

Duas DMEs!

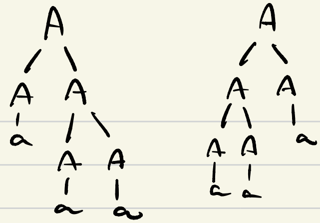
$E \to E + E \to E + E * E \to E + E * t \to E + t * t \to t + t * t$

$E \to E * E \to E * t \to E + E * t \to E + t * t \to t + t * t$

Duas DMDs!

2) $G_2 : A \to AA \mid a$

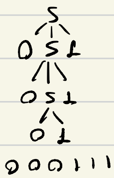
↳ $w = aaa$
(que gera ambiguidade)



$$\begin{cases} A \to AA \to AAA \to aAA \to aaA \to aaa \\ A \to AA \to aA \to aAA \to aaA \to aaa \end{cases}$$

↳ Duas DMEs!

3) $G_3 : S \to 0S1 \mid 01$



} não tem outra forma de gerar essa string

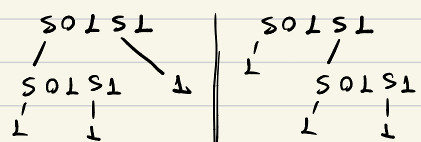
↳ em cada passo da derivação, a forma sentencial gerada sempre será 0^iS1^i, e em todas expansões existe só 1 não-terminal que pode ser expandido.

⇓ ou seja

em cada passo da derivação, não há ambiguidade na escolha de qual variável expandir → força DME e DMD a serem idênticas e únicas

4) $G_4 : S \to S01S1 \mid 1$

↳ gera ambiguidade por sua incapacidade de impor uma <u>hierarquia</u> de agrupamento <u>única</u> p/ as subexpressões



101110111 = 101110111

ADs ≠

⇓

recursão múltipla

(S tem a própria variável S na produção do lado direito de forma repetida. Logo, a gramática falha em especificar <u>qual</u> ocorrência do S é a principal (raiz) na hierarquia)

$a^{2^{K-1}-3}$ | aa | ab | bb | $b^{2^{K-1}-3}$

p q [v] t y

→ NÃO PRECISA FAZER A DIVISÃO EXATA !!!

$a^{2^{K-1}} \mid b^{2^{K-1}} a^{2^{K-1}} \mid b^{2^{K-1}}$

p qvt y

só assim já é suficiente

$ww^Rw \to a^n b^n b^n a^n a^n b^n$

$\Downarrow$

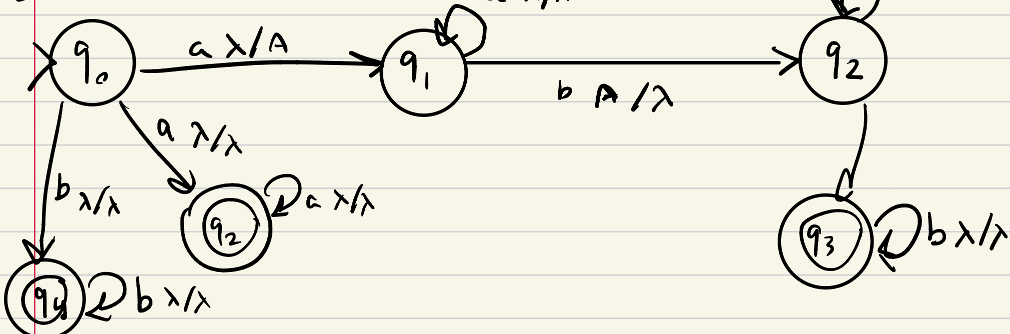
$a^n b^{2n} a^{2n} b^n$

$\Downarrow$

$a^{2^{K-1}} b^{2(2^{K-1})} a^{2(2^{K-1})} b^{2^{K-1}}$

$\left.\begin{matrix} a^n b^n a^n \\ a^n b^n c^n \end{matrix}\right\}$ Não-LLC

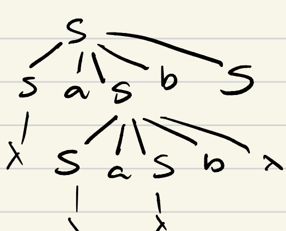
$\{a^n b^m \mid n \neq m\}$



G: S → aSbS | SbSaS | λ

S

⋮ ≠



$a^n b^m \quad n \neq m$

S' → AS | SB

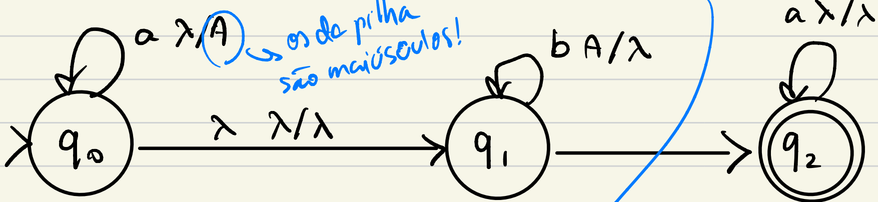
S → aSb | λ

A → aA | a

B → bB | a

$a^n \; b^n \,|\, a^m$ (A = $a^n b^n$, B = a^m)

G: $S \to AB$
$A \to aAb \mid \lambda$
$B \to aB \mid \lambda$



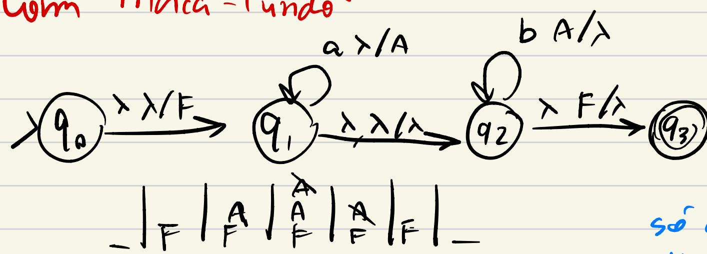
ou

G: $S \to A \mid aSb \mid B$
$A \to aA \mid a$
$B \to bB \mid b$

Linguagem ambígua → Gramática ambígua

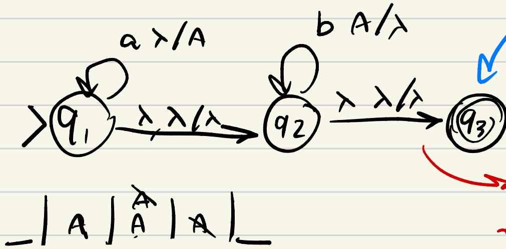
Linguagem ambígua → Gramática não-ambígua
sem uma sentença que deixe ela ambígua

Com Marca-Fundo:



só aceita se:
- pilha estiver vazia
- estar no estado de aceitação

Sem Marca-Fundo:



ele pode até cair no meio de ler os b's, mas ele não aceitaria pq a pilha não tá vazia.

FNC:

$G: S \to A \mid ABa \mid AbA$
$A \to Aa \mid \lambda$
$B \to Bb \mid BC$
$C \to CB \mid CA \mid bB$

$\downarrow$

$S' \to$ "
$G: S \to A \mid ABa \mid AbA \mid \lambda \mid Ba \mid b$
$A \to Aa \mid a$
~~$B \to Bb \mid BC$~~
~~$C \to CB \mid CA \mid bB$~~

B depende de C
C depende de B
⇓
Não produzem terminais
⇓
não acabam nunca

★ REMOVENDO OS INÚTEIS:

quem produz terminais:

$T = \{ S, A .. \}$

$\downarrow$

$S \to Aa \mid a \mid AbA \mid \lambda \mid bA \mid Ab \mid b$
$A \to Aa \mid a$

quem são os inúteis:

- que não produzem terminais
- que não são alcançáveis

Quando crio o S':

se: $S' \to A$
~~$S \to A$~~
} S' e S idênticos e S não tem recursão dele mesmo

⇓

$S' \to A$

removo o inútil que não será atingido (S)

se: $S' \to A \mid S$
$S \to A \mid S$

⇓

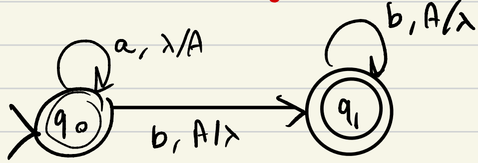
S' e S' não iguais mas S tem recursão nele mesmo (precisa dele!), além dele ser atingível por S'.

1) Construa AP (apenas o diagrama) para:

$$L = \{a^m b^n \mid m \neq n\}$$

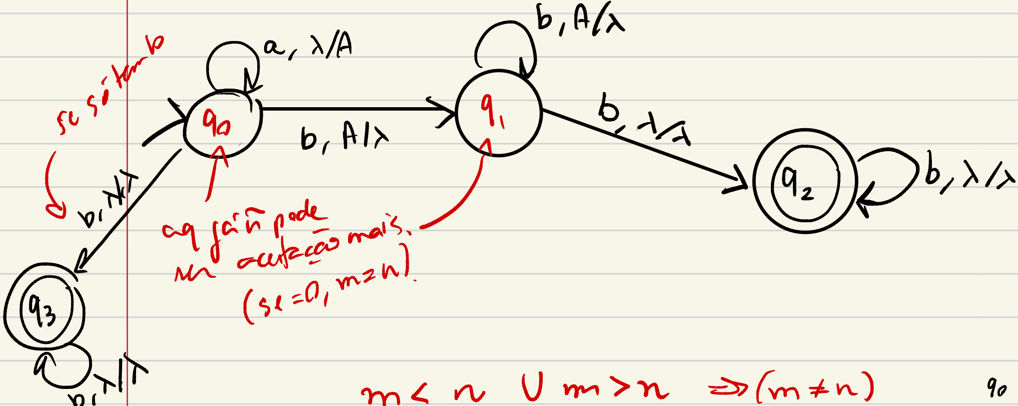
$$L = \{a^m b^n \mid m \neq n\}$$

$m = n$

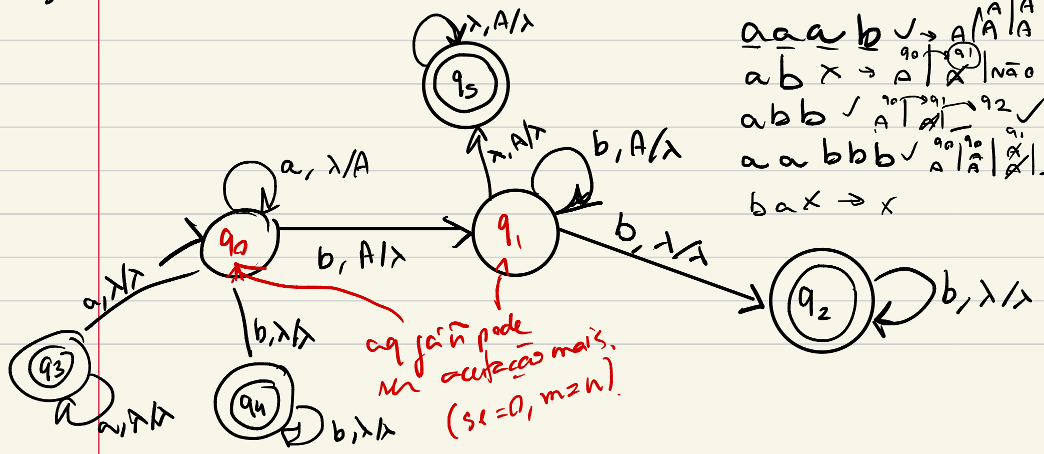


⇓

$m < n$



$m < n \cup m > n \Rightarrow (m \neq n)$



aaab ✓
ab ✗ → não aceitou
abb ✓
aabbb ✓
ba ✗ → ✗

2) Construa um GLC para:

$L = \{a^m b^n \mid m \neq n\}$

$L = \{a^m b^n \mid m \neq n\}$

não pode dar

G: $S \to A \mid B \mid aSb$

$A \to Aa \mid a$

$B \to Bb \mid b$

abb
aaaaab
aaab
aaabb

3) Considere a seguinte GLC: $G = (\{X\}, \{a\}, R, X)$, em que R contém regras: $X \to XX \mid a$. Pede-se:

a) Mostre que G é ambígua

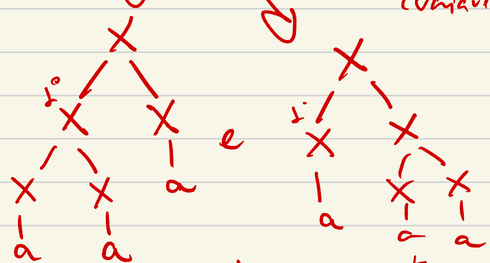
para gerar só 1 a

$X \to XX \mid a$

$G = (\{X\}, \{a\}, R, X)$

variável de partida

(variável) Σ (terminal) produção



ambos DMEs ≠ da mesma string (aaa)

↳ se não fosse ambígua, teria só 1.

ou então:

(mostrando com outro formato de visualização)

2 DMEs { $X \to XX \to aX \to aXX \to aaX \to aaa$

≠ { $X \to XX \to XXX \to aXX \to aaX \to aaa$

b) Forneça uma definição para L(G) utilizando a notação de conjunto

$L = \{ \text{"} \; \}$

expressão $= a^+$

$$L(G) = \{ w \mid w \in \{a\}^+ \}$$

ou

$$L(G) = \{ a^n \mid n \geq 1 \}$$

compostas pelo terminal a.

c) Forneça uma gramática não ambígua equivalente a G

$$G: S \rightarrow Sa \mid a$$

$$\left.\begin{array}{c} S \\ | \\ Sa \\ | \\ Sa \\ | \\ a \end{array}\right\} aaa$$

não tem outro caminho p/ gerar as strings de L(G)

$$G = \{ \{S\}, \{a\}, \{S \rightarrow Sa \mid a\}, S \}$$

Prove que $L = \{[W]W \mid W \in \{a,b\}^*\}$ não é livre de contexto

$(W \neq w!)$

$[\underbrace{W}_{w_1}]\underbrace{W}_{w_2} \longrightarrow w = [\underbrace{a^{2^{K-1}} b^{2^{K-1}}}_{w_1}]\,\underbrace{a^{2^{K-1}} b^{2^{K-1}}}_{w_2}$

1) $w = pqvty$
2) $|q| + |t| > 0$
3) $|qvt| \le 2^{K-1}$
4) $w_i = pq^ivt^iy \in L(G), \forall i \geq 0$

1° CASO w_1 — bombear 1 só tipo de terminal

$[\overbrace{a^{2^{K-1}}}^{w_1} b^{2^{K-1}}]\, a^{2^{K-1}} b^{2^{K-1}}$

$p \mid qvt \mid y$

→ a's de w_1 > a's de w_2

$w_1 \neq w_2$

$w_i \notin L(G)$

→ análogo p/ todos outros tipos de terminal

2° CASO — bombear ~~selecionar~~ simult. 2 terminais

$[a^{2^{K-1}} b^{2^{K-1}}]\, a^{2^{K-1}} b^{2^{K-1}}$

$p \mid qvt \mid y$

→ desordena a ordem de w_1

análogo p/ a b de w_2

impossível p/ a_{w_1} e b_{w_2} $(|qvt| > 2^{K-1})$

3° CASO

$[a^{2^{K-1}} \mid b^{2^{K-1}} \mid] \mid a^{2^{K-1}} \mid b^{2^{K-1}}$

$q \quad v \quad t$

bombear 2 terminais

→ + b's de w_1 mas b's de w_2 =

$w_1 \neq w_2 \rightarrow w_i \notin L(G)$

→ p/ as e bs de w_1 → a's e b's de w_1 +, a's e b's de w_2 =.

→ impossível p/ as de w_1 e bs de w_2 $(|qvt| > 2^{K-1})$

4° CASO

$[a^{2^{K-1}} \mid b^{2^{K-1}} \mid]\, a^{2^{K-1}} \mid b^{2^{K-1}} \mid$

$q \quad v \quad t$ → impossível $(|qvt| > 2^{K-1})$

→ análogo para outros

PROVA POR ABSURDO

5º CASO

$$p \mid \underbrace{[a^{2^{k-1}} b^{2^{k-1}}]}_{qvt} a^{2^{k-1}} b^{2^{k-1}} \mid y \rightarrow \text{impossível } (|qvt| > 2^{k-1})$$

Dada uma gramática livre de contexto, qual das alternativas abaixo caracteriza corretamente uma linguagem inerentemente ambígua?

A É possível criar pelo menos uma gramática não ambígua para essa linguagem.

B Toda gramática que gera essa linguagem será necessariamente ambígua.

C A linguagem é sempre regular.

D A linguagem só pode ser reconhecida por um autômato finito determinístico.

✓ Correto

Exatamente: toda gramática livre de contexto para essa linguagem apresenta ambiguidade.

Sobre as propriedades de fechamento das linguagens livres de contexto (LLC), qual das opções é incorreta?

A LLCs são fechadas por união.

B LLCs são fechadas por interseção geral.

C LLCs são fechadas por concatenação.

D LLCs são fechadas por interseção com linguagens regulares.

✓ Correto

Correto: LLCs não são fechadas sob interseção geral entre duas LLCs.

Segundo o lema do bombeamento para LLC, selecione a afirmação falsa a respeito da string u v w x y e suas condições de bombeamento:

A Há sempre um símbolo não-terminal repetido em algum caminho da árvore de derivação de comprimento suficiente.

B Existe uma string suficientemente longa em L tal que, ao repetir ou remover os blocos v e x, a nova string ainda pertence a L.

C Se uma linguagem não satisfaz as condições do lema do bombeamento, ela certamente não é livre de contexto.

D Toda linguagem que satisfaz o lema do bombeamento é, necessariamente, livre de contexto.

✓ Correto

Esta é falsa: existem linguagens que não são livres de contexto, mas satisfazem o lema. O lema é necessário, mas não suficiente.

Considere as gramáticas G1: S → aSb | ε e G2: S → aSa | bSb | a | b. Sobre ambiguidade, afirmamos: ?

A Ambas são necessariamente ambíguas para a palavra aba.

B A gramática G1 é não ambígua, enquanto G2 pode ser ambígua para algumas cadeias.

C A G2 não pode ser ambígua para nenhuma cadeia da linguagem.

D Ambas nunca geram mais de uma árvore de derivação para a mesma string.

✕ Incorreto

G2 pode gerar mais de uma árvore (ambiguidade), então esta está errada.

1. **Sobre o Lema do Bombeamento para LLC, considere o seguinte:**

Uma linguagem L é LLC gerada por uma gramática na Forma Normal de Chomsky com k símbolos não-terminais. Se uma string $z \in L$ com $|z| \geq 2^{k-1}$, qual das alternativas não é necessariamente verdadeira?

a) z pode ser decomposta como $z = uvwxy$ com $|vwx| \leq 2^{k-1}$ e $|vx| \geq 1$
b) Para todo $i \geq 0$, $uv^iwx^iy \in L$
c) O símbolo que se repete no caminho mais longo da árvore de derivação é um símbolo terminal
d) A decomposição $uvwxy$ garante a possibilidade de bombear sem sair da linguagem

5. **Qual das seguintes gramáticas é necessariamente não ambígua?**

a) $S \to SS \mid a$
b) $S \to aSb \mid \lambda$
c) $S \to aS \mid a$
d) $S \to aSa \mid bSb \mid a \mid b$

7. **Quanto à árvore de derivação e as derivações mais à esquerda (DME) e mais à direita (DMD):**

É correto afirmar:

a) Se a gramática não é ambígua, DME e DMD sempre geram a mesma árvore de derivação
b) DME e DMD são sempre diferentes na mesma gramática
c) A existência de múltiplas DME garante que a gramática seja não ambígua
d) DMD nunca pode ser igual a DME em gramáticas livres de contexto