

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E DE ESTATÍSTICA  
INE5421 – LINGUAGENS FORMAIS E COMPILADORES**

**LISTA DE EXERCÍCIOS N. 4 – 12/1**

1) Construa um PDA P |

- a)  $T(P) = \{x c y \mid x, y \in (a, b)^* \wedge x^r \text{ está contido em } y\}$
- b)  $N(P) = \{a^n b^m \mid n, m \geq 0 \wedge n \neq m\}$
- c)  $T(P) = \{x \mid x \in a^n (b, c)^* \wedge \# b's = n + \# c's\}$
- d)  $N(P) \text{ ou } T(P) = \{a^n b^m \mid n \geq 0, 2n > m \geq n\}$
- e)  $N(P) \text{ ou } T(P) = \{\text{expressões regulares sobre o alfabeto } \{a, b\}\}$

2) Construa o PDA P (com um único estado) |  $N(P) = \{a^n b^m c^m d^n \mid n \geq 0, m \geq 1\}$  e, em seguida, transforme esse PDA em uma GLC equivalente.

3) a) Determine  $T(P)$  onde P é dado por:

$P = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ , onde:

$K = \{q_0, q_1\}, \Sigma = \{0, 1\}, \Gamma = \{Z, X\}, q_0 = q_0, Z_0 = Z, F = \{q_1\}$

$\delta = \{\delta(q_0, 1, Z) = (q_0, XZ)$

$\delta(q_0, 1, X) = (q_0, XX)$

$\delta(q_0, \epsilon, X) = (q_1, X)$

$\delta(q_1, 1, X) = (q_1, \epsilon)$

$\delta(q_1, 0, X) = (q_1, X)$

$\delta(q_1, 0, Z) = (q_1, Z)$

$\delta(q_1, \epsilon, Z) = (q_1, \epsilon)\}$

b) Determine  $N(P)$  onde P é definido por:

$P = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ , onde:

$K = \{q_0, q_1, q_2, q_3\}, \Sigma = \{a, b\}, \Gamma = \{Z, R\}, q_0 = q_0, Z_0 = Z, F = \emptyset$

$\delta = \{\delta(q_0, a, Z) = (q_0, R), \delta(q_1, b, R) = (q_1, RR),$

$\delta(q_0, a, R) = (q_0, RR), \delta(q_1, b, R) = (q_3, \epsilon),$

$\delta(q_0, b, Z) = (q_1, R), \delta(q_1, a, R) = (q_2, \epsilon),$

$\delta(q_0, b, R) = (q_1, RR), \delta(q_2, a, R) = (q_2, \epsilon),$

$\delta(q_0, a, R) = (q_2, \epsilon), \delta(q_2, b, R) = (q_3, \epsilon),$

$\delta(q_0, b, R) = (q_3, \epsilon), \delta(q_3, b, R) = (q_3, \epsilon)\}$

c) Os PDA's dos itens anteriores são não-determinísticos? Em caso positivo, identifique os não-determinismos existentes e construa, se possível, PDA's determinísticos equivalentes aos originais.

4) Proponha algoritmos em passos gerais para:

- a) Transformar um PDA que aceita por Pilha Vazia em um PDA equiv. que aceita por Estado Final;
- b) Transformar um PDA que aceita por Estado Final em um PDA equiv. que aceita por Pilha Vazia;

5) Seja G a seguinte GLC:

**$S \rightarrow a S b \mid b S a \mid ab \mid ba$**

Pede-se:

- a) Usando o algoritmo **SHIFT-REDUCE** com B-T, analise a sentença "abab".
- b) Recursão a Esquerda, Não-Fatoração  $\epsilon$ -Produções e Produções simples causam problema no algoritmo SHIFT-REDUCE proposto? Se sim, como resolvê-los?

c) G (original ou transformada) é analisável por um Parser DESCENDENTE RECURSIVO? Em caso positivo, construa-o e simule a análise de uma sentença; em caso negativo, construa o parser da gramática transformada e identifique os problemas decorrentes de seu uso.

6) Transforme, se possível, a gramática G abaixo especificada em LL(1) e construa o Parser Descendente Recursivo para ela, simulando o reconhecimento de uma sentença. Caso G não seja transformável construa o parser mesmo assim, e comente os problemas (e suas causas) desse parser.

$$\begin{aligned} G: \quad S &\rightarrow c S c \mid B c \\ B &\rightarrow a B \mid B b \mid \epsilon \end{aligned}$$

7) As gramáticas abaixo são LL(1) ou transformáveis? em caso positivo construa o parser LL(1) das gramáticas (originais ou transformadas) e exemplifique o processo de reconhecimento de uma sentença correta e de uma sentença incorreta de cada uma delas. Em caso negativo, construa a TP e indique os conflitos encontrados, justificando-os.

a) 
$$\begin{aligned} P &\rightarrow P ; B \mid B \\ B &\rightarrow K V C \\ K &\rightarrow c K \mid \epsilon \\ V &\rightarrow v V \mid \epsilon \\ C &\rightarrow b K V ; C e \mid b C e \mid C com \mid \epsilon \end{aligned}$$

b) 
$$\begin{aligned} P &\rightarrow D C \\ D &\rightarrow T V \\ T &\rightarrow type T \mid \epsilon \\ V &\rightarrow var V \mid \epsilon \\ C &\rightarrow begin D ; C end \mid com C \mid \epsilon \end{aligned}$$

c) A gramática não-ambígua do IF-THEN-ELSE

d) 
$$\begin{aligned} S &\rightarrow a S \mid R \\ R &\rightarrow a R b \mid \epsilon \end{aligned}$$

e) 
$$\begin{aligned} S &\rightarrow A a A b \mid B b B a \\ A &\rightarrow \epsilon \\ B &\rightarrow \epsilon \end{aligned}$$

8) Construa o Parser Descendente Recursivo da Gramática do item 7b e forneça a sequência de ativações usadas na análise de uma determinada sentença.

9) Construa o parser SLR(1) das gramáticas da questão 7 e simule o reconhecimento de sentenças corretas e incorretas daquelas gramáticas usando o algoritmo SLR(1).

OBS.: Utilize as Gram. originais! Em caso de conflitos, analise a sentença sem efetuar back-track.

10) Transforme a GLC G do item 7b em uma GLC G' na FNG e construa um PDA P a partir de G'.

11) As afirmações abaixo estão corretas? Justifique sua resposta usando exemplos e/ou aspectos conceituais estudados nesta disciplina.

a) Gramáticas ambíguas não são nem LL(1) nem SLR(1).

b) Nem toda GLC não-ambígua é analisável pelas técnicas LL(1) e SLR(1).

c) Toda Linguagem Regular é LL(1) e também SLR(1)

d) Nem toda LLC LL(1) é SLR(1) e vice-versa.