

LISTA DE EXERCÍCIOS N. 4 – 13/2

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)  $T(P) = \{x \mid x \in a^n (b, c)^* \wedge \# b's = n + \# c's\}$
- c)  $N(P)$  ou  $T(P) = \{a^n b^m c^k \mid n, m, k \geq 0 \wedge n = 2 * m \text{ ou } m = 2 * k\}$
- d)  $T(P) = \{x \mid x \in (0, 1)^* \wedge \# 1's = 2 * \# 0's\}$
- e)  $N(P)$  ou  $T(P) = \{a^n b^m \mid n \geq 0, 2n \geq m \geq n\}$
- f)  $N(P)$  ou  $T(P) = L(G)$  onde G é a GLC que representa expressões aritméticas (onde  $V_T = \{id, +, *, (, )\}$ ).
- g) Construa, se possível, PDA's determinísticos para representas as linguagens dos itens “e” e “f”

2) 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:

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

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

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

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

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

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

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

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

d) Construa um PDA P'  $\equiv$  ao PDA P do item b | P' tenha um único estado.

e) Construa uma GLC G |  $L(G) = N(P')$

3) Seja G a seguinte GLC:  $S \rightarrow a S b \mid a a S b \mid a S b b \mid \epsilon$

Pede-se:

- a) Usando o algoritmo **SHIFT-REDUCE** com B-T, analise a sentença “aab”.
- b) O algoritmo SHIFT-REDUCE proposto apresenta problemas na análise de sentenças geradas por GLC com **ciclos**, com  **$\epsilon$ -produções** ou com **Recursão a esquerda**? Se sim, como adaptá-lo para resolver os problemas encontrados ?
- c) Construa um PDA P |  $N(P)$  ou  $T(P) = L(G)$ .
- d) G 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.

4 - 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.

**G:  $C \rightarrow \text{if } E \text{ then } C \text{ else } C \mid \text{if } E \text{ then } C \mid \text{com}$**   
 **$E \rightarrow E \text{ or id} \mid E \text{ and id} \mid \text{id}$**

5 - 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)  **$P \rightarrow P ; B \mid B$**   
 **$B \rightarrow K V C$**   
 **$K \rightarrow c K \mid \epsilon$**   
 **$V \rightarrow v V \mid \epsilon$**   
 **$C \rightarrow b K V ; C e \mid b C e \mid C \text{ com} \mid \epsilon$**

b)  **$S \rightarrow A B$**   
 **$A \rightarrow C D \mid A a$**   
 **$B \rightarrow b A B \mid B e \mid \epsilon$**   
 **$C \rightarrow c C \mid \epsilon$**   
 **$D \rightarrow d D \mid \epsilon$**

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

d)  **$S \rightarrow L = R \mid R$**   
 **$L \rightarrow * R \mid \text{id}$**   
 **$R \rightarrow L$**

e)  **$S \rightarrow A a A b \mid B b B a$**   
 **$A \rightarrow \epsilon$**   
 **$B \rightarrow \epsilon$**

6 - Construa o parser SLR(1) das gramáticas da questão 5 e simule o reconhecimento de sentenças corretas e incorretas daquelas gramáticas usando o algoritmo SLR(1).

OBS.: Utilize as Gramáticas originais!

7 - 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 GLC LL(1) é SLR(1) e vice-versa.
- e) Nas tabelas de parsing LL(1) e SLR(1), a existência de colunas sem nenhuma ação sintática significa que o símbolo correspondente é inútil.
- f) Como obter o “parse” diretamente a partir da análise sintática usando a técnica LL(1)? E usando a técnica SLR(1)?
- g) O fato de uma GLC G não ser LL(1) nem transformável implica que L(G) não é LL(1)?