

LISTA DE EXERCÍCIOS SOBRE COMPILADORES

1) Considere a seguinte gramática:

$$\begin{aligned} E &\rightarrow E + E \mid E - T \mid T \\ E &\rightarrow T * F \mid T / F \mid F \\ F &\rightarrow (E) \mid \text{id} \end{aligned}$$

- i) Calcule o FIRST e FOLLOW para a gramática.
- ii) Elabore a tabela de parsing do analisador sintático preditivo para esta gramática.

2) Considere a gramática

$$\begin{aligned} \text{expr} &\rightarrow \text{expr } \mathbf{or} \text{ termo} \mid \text{termo} \\ \text{termo} &\rightarrow \text{termo } \mathbf{and} \text{ fator} \mid \text{fator} \\ \text{fator} &\rightarrow \mathbf{not} \text{ fator} \mid (\text{expr}) \mid \mathbf{true} \mid \mathbf{false} \end{aligned}$$

- i) Construa uma árvore gramatical (parsing) para a seguinte sentença:
not (true or false and true).
- ii) Esta gramática é ambígua? Justifique sua resposta.

Considere a seguinte gramática para os exercícios 3 e 4.

$$\begin{aligned} S &\rightarrow (L) \mid a \\ L &\rightarrow L, S \mid S \end{aligned}$$

- 3) Determine a deviração mais à esquerda para cada uma das seguintes sentenças:
- i) (a,a)
 - ii) (a, ((a,a), (a,a)))

- 4) Determine a deviração mais à direita para cada uma das seguintes sentenças:
- i) (a,a)
 - ii) (a, (a,a))

5) Encontre o nullable, FIRST e FOLLOW dos seguintes conjuntos gramaticais:

- i) $S' \rightarrow S \$$
- ii) $S \rightarrow$
- iii) $S \rightarrow X S$
- iv) $B \rightarrow \backslash \text{begin} \{ \text{WORD} \}$
- v) $B \rightarrow \backslash \text{end} \{ \text{WORD} \}$
- vi) $X \rightarrow B S E$
- vii) $X \rightarrow \{ S \}$
- viii) $X \rightarrow \text{WORD}$
- ix) $X \rightarrow \text{begin}$
- x) $X \rightarrow \text{end}$
- xi) $X \rightarrow \backslash \text{WORD}$

6) Construa a tabela parsing LL(1) utilizando os resultados obtidos no exercício 5.

7) Utilizando a seguinte gramática, construa os estados LR(0) e determine se é uma gramática SLR.

i) $S \rightarrow B \$$

ii) $B \rightarrow id P$

iii) $B \rightarrow id (E]$

iv) $P \rightarrow$

v) $P \rightarrow (E)$

vi) $E \rightarrow B$

vii) $E \rightarrow B, E$

8) A seguinte gramática é posta à fim de remediar a ambiguidade do *else-vazio*, supondo o enunciado *if-then-else*. Mostre que esta gramática ainda é ambígua.

$cmd \rightarrow \mathbf{if\ expr\ then\ cmd}$
 | $cmd_associado$

$cmd_associado \rightarrow \mathbf{if\ expr\ then\ cmd_associado\ else\ cmd}$
 | \mathbf{outro}

9) Construa uma tabela sintática SLR utilizando a seguinte gramática:

$E \rightarrow E + T \mid T$

$T \rightarrow T F \mid F$

$F \rightarrow F * \mid a \mid b$

10) Demonstre que a gramática abaixo é LL(1) mas não é SLR(1).

$S \rightarrow AaAb \mid BbBa$

$A \rightarrow \epsilon$

$B \rightarrow \epsilon$

11) Escreva um programa Yacc que irá receber expressões aritméticas como entrada e produzir como saídas as expressões pos-fixas correspondentes.

12) Traduza cada uma das expressões regulares em gramática livre de contexto.

i) $((xy*x) \mid (yx*y)) ?$

ii) $((0 \mid 1)^+ \text{“.”}(0 \mid 1)^*) \mid ((0 \mid 1)^*\text{“.”}(0 \mid 1)^+)$