

**Construção de Compiladores 1 - 2018.1 - Profs. Mário César San Felice
(e Helena Caseli, Murilo Naldi, Daniel Lucrédio)
Tópico 04 - Análise Sintática Descendente - Lista de Exercícios**

1. Qual a diferença conceitual entre análise sintática ascendente e descendente?
2. Quais as vantagens do analisador preditivo com relação ao analisador com retrocesso? Existe alguma desvantagem?
3. Qual o principal desafio do analisador sintático preditivo?
4. Construa os conjuntos Primeiros e Seguidores para as seguintes gramáticas:

a) S : bAb
A : CB | a
B : Aa
C : c | ϵ

b) S : A | B | ϵ
A : A + B | A - B | 1 | 2 | 3 | ϵ
B : A | C
C : (A)

c) S : ϵ | abA | abB | abC
A : aSaa | b
B := bSbb | c
C := cScc | d

d) E : TE'
E' : +TE' | ϵ
T : FT'
T' : *FT' | ϵ
F : (E) | id

e) declaração : ifdecl | "outra"
ifdecl : "if" "(" exp ")" declaração elseparte
elseparte : "else" declaração | ϵ
exp : "0" | "1"

5. Dada a gramática a seguir:

Expr : Expr 'OU' Termo | Termo
Termo : Termo 'E' Fator | Fator
Fator : 'NÃO' Fator | id

- a) Ela é LL(1)? Se não, aplique as transformações necessárias para convertê-la para LL(1).
- b) Construa a tabela sintática LL(1) correspondente à gramática (alterada na letra a) se for o caso).

6. Considere a gramática

lexp : atomo | lista
atomo : numero | identificador
lista : (lexpseq)
lexpseq : lexpseq lexp | lexp

- a) Remova a recursão à esquerda.
- b) Construa os conjuntos Primeiros e Seguidores para os não-terminais da gramática resultante (letra a).
- c) Construa a tabela de análise sintática a ser usada por um método de ASD preditiva não recursiva, a partir da gramática resultante (letra a).
- d) Mostre as ações do analisador preditivo não recursivo correspondente (de acordo com a tabela criada na letra c) dada a cadeia de entrada (x (y (2)) (z)). OBS.: x, y e z são identificadores e 2, número.
- e) Repita o exercício d) para a cadeia de entrada (x y 2)) (z)). OBS.: x, y e z são identificadores e 2, número.
- f) Repita o exercício d) para a cadeia de entrada (x y 2. OBS.: x, y e z são identificadores e 2, número.

7. A gramática a seguir é LL(k).

```
S : id ':' id | id ':' id '{' S '}' ;
```

Qual o valor de k?

8. A gramática a seguir é LL(k)? Justifique sua resposta

```
declaracao : nomeQualificado ':' ID ';'
           | nomeQualificado ':' ID '=' expressao;
nomeQualificado : ID | ID '.' nomeQualificado;
```

9. O que é um DFA de decisão?

10. Qual a diferença entre DFAs de decisão para gramáticas LL(k) e para gramáticas LL(*)?

11. Porque a gramática do exercício 8 não é LL(*)? Como fazer com que ela seja LL(*) sem modificar a linguagem?

12. Classifique as seguintes gramáticas como sendo não-LL, LL(1), LL(k) ou LL(*)

a)

```
S : A | B | ε
A : A + B | A - B | 1 | 2 | 3 | ε
B : A | C
C : (A)
```

b)

```
S : A | B | ε
A : + A B | - A B | 1 | 2 | 3 | ε
B : A | C
C : (A)
```

c)

```
S : + A B | + A C | + A d
A : (-|/)*
B : [S]
C : (S)
```

d)

```
S : + A + | + A - | + A / | + A *
A : id | num | (S)
```