

Questões

Questão 1. [1 PONTO]

Explique detalhadamente *o que é e como é construído* um **gerador de analisador léxico**.

Questão 2. [1 PONTO]

Escreva uma expressão regular que define a linguagem dos **literais em ponto flutuante**.

Um literal em ponto flutuante pode ter um sinal opcional, seguida da mantissa, que é descrita por uma sequência de dígitos decimais seguida por um ponto decimal e então por outra sequência de dígitos. Uma e apenas uma dessas sequências de dígitos pode ser vazia. Finalmente há uma parte exponencial opcional, formada pela letra *e* maiúscula ou minúscula, seguida por um sinal opcional e por uma sequência não vazia de dígitos decimais. Se houver uma parte exponencial, a parte da mantissa pode ser escrita sem o ponto decimal.

Exemplos:

3.14

-3.

.23

3e4

11.22E-3

0.17e+34

Questão 3. [1 PONTO]

Considere a cadeia `abbbbaacc`. Quais das seguintes especificações léxicas produzem a sequência de *tokens* `ab, bb, a, acc`? (Marque todas as respostas que se aplicam.)

1. `ab`
`b+`
`ac*`

2. `b+`
`ab*`
`ac*`

3. `c*`
`b+`
`ab`
`ac*`

4. `a(b|c*)`
`b+`

Questão 4. [1 PONTO]

Considere a seguinte especificação léxica:

`(00)*`

`01+`

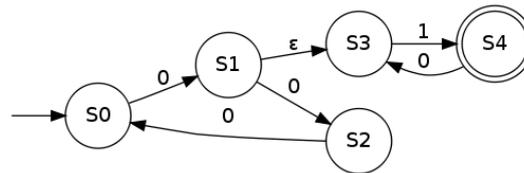
`10+`

Que cadeias **não** são processadas com sucesso por esta especificação? (Marque todas que se aplicam.)

1. 01100110
2. 0001101
3. 0111110
4. 01100100

Questão 5. [1 PONTO]

Quais das seguintes expressões regulares geram a mesma linguagem que é reconhecida pelo autômato finito não determinístico abaixo? (Marque todas as respostas que se aplicam.)



1. $(000)^*(01)^+$
2. $0(00)^*(10)^*$
3. $(000)^*(10)^+$
4. $0(000)^*(01)^*$
5. $0(000)^*1(01)^*$

Questão 6. [2 PONTOS]

Dada a gramática

$$\begin{aligned}
 A &\rightarrow - A \\
 A &\rightarrow A - id \\
 A &\rightarrow id
 \end{aligned}$$

1. Mostre que esta gramática é ambígua encontrando uma cadeia da linguagem que tenha duas árvores sintáticas diferentes.
2. Escreva uma gramática não ambígua para a mesma linguagem onde o operador prefixo *menos* tem prioridade sobre o operador infix *menos*.
3. Escreva uma gramática não ambígua para a mesma linguagem onde o operador infix *menos* tem prioridade sobre o operador prefixo *menos*.
4. Mostre as árvores sintáticas usando as novas gramáticas para a cadeia que você usou para provar que a gramática original é ambígua.

Questão 7. [3 PONTOS]

Dada a gramática livre de contexto

```
S → E ;
S → id := E ;
S → if E then S else S
S → while E do S
S → { L }
L → S L
L →
E → A == A
E → A
A → T + A
A → T
T → F * T
T → F
F → id
F → ( E )
```

1. Determine quais símbolos não terminais da gramática são *nullable*.
2. Determine o conjunto *FIRST* de cada símbolo não terminal da gramática.
3. Determine o conjunto *FOLLOW* de cada símbolo não terminal da gramática.