

Lista de Linguagens de Programação – 2

Nome: _____ Matrícula: _____

1. Esta questão refere-se à gramática abaixo:

```
<frase>          ::= <expr_nominal> <predicado>
<expr_nominal> ::= <artigo> <nome>
                  | <artigo> <nome> <expr_propos>
<predicado>      ::= <verbo>
                  | <verbo> <expr_nominal>
                  | <verbo> <expr_nominal> <expr_propos>
<expr_prepos>    ::= <preposicao> <expr_nominal>
<nome>           ::= "menino" | "menina" | "pato" | "telescopio"
                  | "musica" | "pena"
<preposicao>      ::= "com" | "ate"
<verbo>          ::= "viu" | "esta" | "e" | "canta"
                  | "surpreende" | "toca"
<artigo>         ::= "um" | "uma" | "o" | "a"
```

- (a) Uma gramática é ambígua quando uma mesma sentença possui duas ou mais árvores de derivação diferentes. Construa duas árvores de derivação distintas para a seguinte sentença: “A menina toca o pato com a pena”.
- (b) Se considerarmos que cada palavra da gramática possui o significado dado por um dicionário de Português, quais seriam as duas interpretações possíveis para a frase “A menina toca o pato com a pena”?

2. Neste exercício vocês desenvolverão uma gramática lógica em Prolog, e verão que trabalhar com gramáticas em Prolog é bem fácil. Existem interpretadores Prolog instalados nas máquinas das salas 2019 e 2020. Por exemplo, `sirius.grad`.

(a) Construa uma gramática lógica, em Prolog, para a gramática vista na questão anterior, e a transcreva abaixo:

(b) Como saber que ‘a,menina,toca,o,pato,com,a,pena’ possui duas derivações, enquanto ‘a,menina,toca,o,pato’ possui somente uma?

(c) Quantas soluções existem para a seguinte busca:
`frase([a,Sujeito,toca,o,pato],[]).`

3. Considere as duas gramáticas abaixo, e diga se alguma delas é ambígua. O símbolo ϵ significa a palavra vazia. Estas gramáticas reconhecem a mesma linguagem, formada por *strings* enclausuradas por parênteses e colchetes balanceados. Se a gramática for ambígua, escreva as suas duas árvores de derivação. Do contrário explique porque você acha que a gramática não apresenta ambiguidades.

(a) Primeira gramática:

```
<string> ::= <string> <string>
           | ( <string> )
           | [ <string> ]
           |  $\epsilon$ 
```

(b) Segunda gramática:

```
<string> ::= ( <string> ) <string>
           | [ <string> ] <string>
           |  $\epsilon$ 
```

4. Existe uma variante das gramáticas formais chamada *Sistemas L*, em homenagem ao botânico Aristid lindenmayer, que popularizou o conceito. Um sistema L é formado por uma quadrupla $L = (V, T, \omega, P)$, tal que:

- V é um conjunto de símbolos não terminais.
- T é um conjunto de terminais.
- ω é um axioma, ou string inicial.
- P é um conjunto de regras de produção.

Porém, ao contrário de nossas gramáticas, as regras de produção em um sistema L são aplicadas todas de uma vez. Por exemplo, abaixo temos um sistema L que produz strings sempre com o tamanho dos números da sequência de Fibonacci:

variáveis: $\{A, B\}$

terminais: $\{\}$

axioma: A

regras: $A \rightarrow B, B \rightarrow AB$

Temos as seguintes strings, para diferentes números de vezes que as regras de produção foram aplicadas:

$n = 0 : A$

$n = 1 : B$

$n = 2 : AB$

$n = 3 : BAB$

$n = 5 : BABABBAB$

$n = 6 : ABBABBABABBAB$

$n = 7 : BABABBABABBABABBABABBAB$

Responda com sinceridade: a relação de tamanho entre a string n e a string $n - 1$ é visualmente agradável? Os cavalheiros que construíram o *Paternon* achavam que sim...

Enfim... o legal mesmo a respeito de sistemas L é quando damos uma interpretação geométrica aos terminais e não terminais. Imagine que você possa amarrar uma caneta à cauda de um exemplar da rara Tartaruga da Patália. A tartaruga, então, é posta para analisar, sequencialmente, uma string produzida por um sistema L. Sempre que ela vê um não terminal, ela caminha dez passinhos, marcando o chão, de papel, com a sua caneta. Sempre que ela vê um terminal ela dá uma guinada, cujo ângulo é uma constante associada ao não terminal. Se você quiser saber mais sobre o assunto, note que sistemas L são um tipo de *fractal*.

Nesta questão faremos o experimento da tartaruga, para as três primeiras iterações de diferentes sistemas L:

- (a) Um $+$ significa que a tartaruga deve virar 90 graus. Um $-$ comanda a pobre a virar 270:

variáveis: $\{F\}$

terminais: $\{+, -\}$

axioma: F

regras: $F \rightarrow F + F - F - F + F$

- (b) Um $+$ manda a tartaruga virar 60 graus. Um $-$ comanda a miserável a virar 300:

variáveis: $\{A, B\}$

terminais: $\{+, -\}$

axioma: A

regras: $A \rightarrow B - A - B, B \rightarrow A + B + A$

- (c) Mesma interpretação que na questão anterior, isto é: um $+$ significa que a nossa *turtle* deve virar 60 graus. Um $-$ a comanda a virar 300:

variáveis: $\{F\}$

terminais: $\{+, -\}$

axioma: $F + +F + +F$

regras: $F \rightarrow F - F + +F - F$