Lista de Exercícios 01
Sintaxe e Semântica

Exercício 01) Considere a gramática a seguir para construção de frases em língua portuguesa.

```
<frase>         ::= <expr_nominal> <predicado>
<expr_nominal>  ::= <artigo> <nome>
                  |  <artigo> <nome> <expr_propos>
<predicado>     ::= <verbo>
                  |  <verbo> <expr_nominal>
                  |  <verbo> <expr_nominal> <expr_propos>
<expr_propos>   ::= <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”?

Exercício 02) As questões a seguir são relativas à gramática da questão anterior e a linguagem lógica Prolog.

a) Escreva essa gramática em um programa em Prolog usando a sintaxe de definição de regras com o operador `-->`.
b) Como saber que a frase `[a,menina,toca,o,pato,com,a,pena]` possui duas derivações, enquanto a frase `[a,menina,toca,o,pato]` possui somente uma derivação em Prolog?
c) Quantas soluções existem para a seguinte busca em Prolog?

```
?- frase([a,Sujeito,toca,o,pato],[]).
```

Exercício 03) Insira parênteses no comando abaixo, de acordo com a precedência dos operadores da linguagem C.

```
a = b < c ? * p + b * c : 1 << d ()
```

Exercício 04) Esta questão se refere a gramática abaixo que representa uma linguagem muito simples: somas de números. Por simplicidade não serão mostradas as regras de produção para os números.

```
⟨E⟩ ::= ⟨E⟩ + ⟨E⟩
      | ⟨Number⟩
```

a) Prove que a gramática em questão é ambígua.
b) Mostre como esta ambiguidade compromete a semântica da linguagem que a gramática representa.
c) Forneça uma gramática que reconheça a mesma linguagem, mas que não seja ambígua.

Exercício 05) Considere as duas gramáticas a seguir e diga se alguma delas é ambígua. A regra `<empty>` não gera símbolos. Ambas gramáticas reconhecem a mesma linguagem: *strings* formadas por parênteses e colchetes balanceados. Se a gramática for ambígua, mostre suas duas árvores de derivação. Caso contrário, explique porque você acha que a gramática não apresenta ambiguidades.

a) Primeira gramática.

```
<string> ::= <string> <string>
           | '(' <string> ')'
           | '[' <string> ']'
           | <empty>
```

b) Segunda gramática.

```
<string> ::= '(' <string> ')' <string>
           | '[' <string> ']' <string>
           | <empty>
```

Exercício 06) As questões a seguir referem-se ao conceito de associatividade de operadores.

a) O que significa dizer que um operador é associativo à esquerda ou à direita?
b) Considere o operador de soma aritmética +, usado na linguagem C. Este operador é associativo à esquerda ou à direita?
c) Dê um exemplo de um operador em C que seja associativo à direita.
d) Modifique a gramática abaixo para que operadores sejam associativos à esquerda:

```
<exp>     ::= <mulexp> + <exp>
            |  <mulexp>
<mulexp>  ::= <rootexp> * <mulexp>
            |  <rootexp>
<rootexp> ::= ( <exp> )
            |  <number>
```

Exercício 07) As questões a seguir são relativas à gramática da questão anterior e a linguagem lógica Prolog.

a) Escreva essa gramática em um programa em Prolog usando a sintaxe de definição de regras com o operador `-->`.
b) Modifique a gramática construída no exercício anterior para que ela compute o valor da expressão aritmética. Por exemplo:

```
?- expr(N, [2,*,'(',3,+,4,')'], []).
N = 14 .
```