

	Curso	Ciência da Computação
	Atividade Acadêmica	Tradutores
	Ano/Semestre	2019/02
	Professor	Leandro Teodoro
	Data de Entrega	21/11/2019
	<b>TRABALHO Final – Tradutores</b>	

#### - Gramática 1

Você foi encarregado de implementar um tradutor (analisador sintático e semântico) para uma linguagem que será utilizada para controlar uma sonda que será enviada a outro planeta. A linguagem é bastante simples e é definida pela seguinte gramática:

**comando -> comando ENTAO comando**  
**comando -> comando APOS comando**  
**comando -> (comando)**  
**comando -> basico**  
**basico -> FRENTE n**  
**basico -> ESQUERDA n**  
**basico -> DIREITA n**  
**basico -> TRAS n**

Onde *n* é qualquer número positivo. A sonda estará sempre direcionada para o norte, sul, leste ou oeste. O significado dos comandos (tokens de entrada) são apresentados na tabela abaixo:

Comando	Significado
<i>c1</i> ENTAO <i>c2</i>	Executa <i>c1</i> seguido por <i>c2</i>
<i>c1</i> APOS <i>c2</i>	Executa <i>c2</i> seguido por <i>c1</i>
FRENTE <i>n</i>	Anda para frente <i>n</i> passos
ESQUERDA <i>n</i>	Vira 90 graus para esquerda e anda <i>n</i> passos
DIREITA <i>n</i>	Vira a direita 90 graus e anda <i>n</i> passos
TRAS <i>n</i>	Vira 180 graus e anda <i>n</i> passos

Assuma que a sonda sempre iniciará sua trajetória na posição **0,0**. Escreva um tradutor que, ao receber como entrada uma sequência de comandos, mostre na tela a distância percorrida pela sonda após a execução destes comandos.

#### Exemplo de uma sequência de comandos:

FRENTE 10 ENTAO ESQUERDA 20  
 DIREITA 30 APOS ESQUERDA 20  
 FRENTE 70  
 DIREITA 100  
 TRAS 20

Além disso, considere que cada palavra (ou seja, cada comando: FRENTE, DIREITA, etc) que não fizer parte da gramática desta linguagem, deverá ser identificado e apresentado em que parte

do processo de derivação este erro foi encontrado. Portanto, deverá ser realizada a análise sintática e semântica.

- **Gramática 2**

Construa uma GLC que represente a seguinte linguagem de programação:

\* **Programa:** uma lista de um ou mais comandos. Se houver mais de um comando na lista, eles serão separados por “;”.

\* **Comando de atribuição:** variável seguida de “:=” e de uma expressão aritmética.

\* **Comandos de teste, com duas formas distintas:**

(a) “if” seguido de expressão relacional, seguida de “then”, seguido de lista de um ou mais comandos separados por “;” ou

(b) “if” seguido de expressão relacional, seguida de “then”, seguido de lista de um ou mais comandos separados por “;” finalizada por “else” que é seguido por uma nova lista de um ou mais comandos separados por “;”.

\* **Comando de iteração:** “while” seguido de expressão relacional, seguida de “do”, seguido de uma lista de um ou mais comandos separados por “;”.

\* **Expressão aritmética:** formada com os operadores de “+”, “-”, “\*”, “/” e (opcionalmente) parênteses balanceados aplicados sobre variáveis, constantes ou expressões aritméticas.

\* **Expressão relacional:** formada por uma expressão aritmética seguida de um dos seguintes operadores relacionais: “=”, “<”, “<=”, “>”, “>=”, seguida de outra expressão aritmética.

\* **Variáveis** são palavras de {a,b}\* e as **constantes** são palavras de {0,1}\*.

Assim como para a Gramática, considere que cada palavra (ou seja, trechos de código) que não fizer parte da gramática desta linguagem, deverá ser identificado e apresentado em que parte do processo de derivação este erro foi encontrado. Portanto, deverá ser realizada a análise sintática e semântica.

**Exemplo de um trecho de código:**

```
b := 4 / (2+3);
a := b + c * 1;
if a=1 then
    if b=0 then
        a:=0;
    else b:=1;
```

As duas gramáticas **deverão** ser desenvolvidas utilizando a **ferramenta ANTLR**.

*Bom trabalho, pessoal!!*