

Estrutura da Gramática

A gramática apresentada no código segue a estrutura de **linguagem de montagem** (assembly), sendo dividida em **duas seções principais**: **.DATA** e **.CODE**.

1. Seção de Dados: **.DATA**

A seção **.DATA** é responsável pela **declaração e inicialização** de variáveis e constantes utilizadas no programa. Os principais elementos dessa seção são:

- **Definição de Variáveis e Constantes:**
 - **Variáveis** podem ser declaradas sem valor inicial (ex: **A DB ?** e **RES DB ?**), o que significa que seus valores serão definidos mais tarde no programa.
 - **Constantes** são definidas com valores específicos (ex: **CONST5 DB 5**, **CONST2 DB 2**), e seus valores não mudam durante a execução.
- **Sintaxe:**
 - A palavra-chave **.DATA** indica o início da seção de dados.
 - O comando **DB** (Define Byte) aloca 1 byte de memória para armazenar dados.

Exemplo:

```
.DATA  
A DB ?  
RES DB ?  
CONST5 DB 5  
CONST2 DB 2  
CONST3 DB 3
```

Estrutura:

```
.DATA  
<nome> DB <valor ou ?>
```

Onde:

- **<nome>** é o identificador da variável ou constante.

- **DB** é a diretiva que indica a alocação de um byte de memória.
- **<valor>** é o valor inicial (se fornecido), ou **?** quando o valor será definido em outro momento.

2. Seção de Código: **.CODE**

A seção **.CODE** contém as **instruções** que a máquina deve executar. Essas instruções manipulam os dados definidos na seção **.DATA** e realizam as operações solicitadas pelo programa.

- **Instruções de Execução:**
 - O código é composto por instruções simples, como **LDA** (load), **ADD** (addition), **STA** (store), que são executadas diretamente pela máquina.
 - As instruções podem acessar e modificar as variáveis definidas na seção de dados.
- **Diretivas:**
 - **.ORG 0:** Define o endereço de memória inicial do código.
- **Sintaxe:**
 - As instruções são compostas por comandos abreviados, como:
 - **LDA:** Carrega um valor para o acumulador.
 - **ADD:** Soma um valor ao acumulador.
 - **STA:** Armazena o valor do acumulador em uma variável.
 - **HLT:** Finaliza a execução do programa.

Exemplo:

```
.CODE
.ORG 0
LDA CONST5
ADD CONST2
STA A
LDA A
ADD CONST3
STA RES
HLT
```

Estrutura:

```
.CODE  
.ORG <endereço>  
<instrução> <argumento>
```

Onde:

- **.CODE** inicia a seção de código.
- **.ORG** define o endereço inicial onde o código será carregado na memória.
- **<instrução>** é a operação a ser realizada (ex: **LDA**, **ADD**).
- **<argumento>** é o dado ou variável envolvido na operação.