

## TRABALHO n<sup>o</sup> 3: especificação da gramática livre de contexto

Considere a forma geral de um programa e dos comandos de atribuição, de entrada de dados, de saída de dados, de seleção, de repetição e a declaração de constantes e variáveis da linguagem 2021.1 especificados pela sua equipe.

Fazendo uso da **notação BNF**, construa uma GLC que defina todas as regras sintáticas necessárias para escrever um programa nessa linguagem.

Para a gramática especificada, elimine as recursões à esquerda e os não-determinismos à esquerda dos não-terminais para os quais isso é possível.

**DATA LIMITE:** trabalho a ser desenvolvido em sala de aula

**ENTREGAR:** além das identificações solicitadas, a gramática modificada (sem recursão à esquerda e não-determinismo à esquerda).

**TRABALHO EM EQUIPE**

### EXEMPLO

```
<programa> ::= int main ( ) { <declaração de constantes e variáveis> ...  
<declaração de constantes e variáveis> ::= ...
```

# ESPECIFICAÇÃO DA LINGUAGEM 2021.1

## Forma geral de um programa

```
:- comentário  
program {  
    <declaração de constantes e variáveis>  
    <corpo do programa>  
}  
identificador
```

- *:- comentário* é facultativo
- *comentário* corresponde a um comentário (constante literal) acerca do programa.
- *identificador* corresponde ao identificador do programa e é opcional

## Forma geral da declaração de constantes e variáveis

```
define {  
    not variable  
        <tipo> is <lista de identificadores> <valor> .  
    variable  
        <tipo> is <lista de identificadores> .  
}
```

- <tipo> **is** <lista de identificadores> <valor> . e  
 <tipo> **is** <lista de identificadores> . podem ocorrer uma ou mais vezes;
- <tipo> pode ser **natural**, **real**, **char** ou **boolean**;
- em <lista de identificadores> deve existir no mínimo um identificador e, caso existam mais identificadores, os mesmos serão separados uns dos outros por uma vírgula ( , ); no caso da declaração de variáveis, cada identificador poderá ser seguido por [ *constante numérica inteira* ], indicando uma variável indexada unidimensional, cujos índices variam no intervalo de 1 até a constante numérica especificada;
- <valor> pode ser um valor inteiro, real ou literal, compatíveis com os tipos **natural**, **real** e **char**, respectivamente;
- a declaração de constantes é precedida de **not variable**;
- a declaração de variáveis é precedida de **variable**;
- a declaração de constantes pode preceder a declaração de variáveis ou a declaração de variáveis pode preceder a declaração de constantes e isto ocorrerá apenas uma única vez;
- a declaração de constantes pode não existir, caso não seja utilizada nenhuma constante no programa;
- a declaração de variáveis pode não existir, caso não seja utilizada nenhuma variável no programa;
- a declaração de constantes e variáveis pode não existir, caso não seja utilizada nenhuma constante ou variável no programa.

## Forma geral do corpo do programa

```
execute {  
    <lista de comandos>  
}
```

- em <lista de comandos> deve existir no mínimo um comando.

## Forma geral do comando de atribuição

```
set <expressão> to <lista de identificadores> .
```

- em <lista de identificadores> deve existir no mínimo um e caso existam mais identificadores de variáveis, os mesmos estão separados uns dos outros por uma vírgula ( , );
- a <lista de identificadores> deve se referir a identificadores de variáveis;
- <expressão> pode ser qualquer expressão aritmética, relacional ou lógica envolvendo identificadores e/ou constantes do tipo **natural**, **real**, **char** ou **boolean**;
- o resultado da avaliação de <expressão> deve ser um valor do mesmo tipo (ou de tipo compatível) da lista de identificadores.

#### Forma geral do comando de entrada de dados

**get { <lista de identificadores> } .**

- em <lista de identificadores> deve existir no mínimo um e, caso existam mais identificadores de variáveis, os mesmos serão separados uns dos outros por uma vírgula ( , ).

#### Forma geral do comando de saída de dados

**put { <lista de identificadores e/ou constantes> } .**

- em <lista de identificadores e/ou constantes> deve existir no mínimo um identificador de constante/variável ou uma constante (numérica ou literal) e, caso existam mais identificadores de constantes/variáveis e/ou constantes, os mesmos serão separados uns dos outros por uma vírgula ( , ).
- as constantes e os conteúdos dos identificadores serão apresentados no dispositivo padrão de saída.

#### Forma geral do comando de seleção

**verify <expressão>**  
**is true { <lista de comandos> }**  
**is false { <lista de comandos> } .**

- <expressão> pode ser qualquer expressão relacional ou lógica envolvendo identificadores e/ou constantes do tipo **natural, real, char** ou **boolean**;
- o resultado da avaliação de <expressão> deve ser um valor lógico (**true** ou **false**);
- as cláusulas **is true** e **is false** só podem ocorrer uma vez, em qualquer ordem e são opcionais, mas deve existir no mínimo uma delas.

#### Forma geral do comando de repetição

**loop**  
**{ <lista de comandos> }**  
**while <expressão> is true .**

**while <expressão> is true do**  
**{ <lista de comandos> } .**

- <expressão> pode ser qualquer expressão relacional ou lógica envolvendo identificadores e/ou constantes do tipo **natural, real, char** ou **boolean**;
- o resultado da avaliação de <expressão> deve ser um valor lógico (**true** ou **false**);
- os comandos da estrutura de repetição serão repetidos sempre que o resultado da avaliação da expressão for **true**.

#### Operadores:

- a) aritméticos: + - \* / \*\* (potência) % (divisão inteira) %% (resto da divisão inteira)
- b) relacionais: == (igual), != (diferente), < (menor), > (maior), <= (menor igual) e >= (maior igual)
- c) lógicos: & (e), | (ou) e ! (não)

Podem ser usados para agrupar as expressões aritméticas, relacionais ou lógicas os parênteses ( e ).

São constantes lógicas: **true** (verdadeiro) e **false** (representando falso).

As palavras reservadas podem ser escritas com letras minúsculas e/ou maiúsculas.

Todos os comandos são finalizados com um ponto.