

TRABALHO nº 1

especificação das expressões e definições regulares especificação dos autômatos finitos determinísticos mínimos (linguagem 2021.1)

QUESTÃO 1. Para a forma geral dos comandos e de um programa a ser escrito na linguagem 2021.1, liste as **palavras reservadas** e os **símbolos especiais**.

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 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:

- aritméticos: + - * / ** (potência) % (divisão inteira) %% (resto da divisão inteira)
- relacionais: == (igual), != (diferente), < (menor), > (maior), <= (menor igual) e >= (maior igual)
- 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.

QUESTÃO 2. Especifique um **comando de seleção** para a linguagem 2021.1 e apresente a sua forma geral, descrevendo o seu funcionamento no padrão de especificação estabelecido.

QUESTÃO 3. Sejam $V_L = \{A, B, \dots, Z, a, b, \dots, z\}$ e $V_D = \{0, 1, \dots, 9\}$. Escreva **definições regulares** para as linguagens regulares denotadas pelos *tokens* apresentados na sequência, conforme exemplo:

EXEMPLO:

identificadores:

linguagem: $\{ x \mid x \in (V_L \cup V_D)^+ \wedge 1 \leq |x| \leq 2 \wedge x \text{ começa com uma letra seguida ou não por outra letra ou um dígito} \}$.

OU $\{x \mid x \in (V_L \cup V_D)^+ \wedge x \text{ é composta por qualquer combinação de letras ou dígitos, com tamanho maior ou igual a 1 e menor ou igual 2, iniciando com uma letra }\}$.

definição regular:

ASCII $\Rightarrow \#0 \mid \dots \mid \#255$

letras $\Rightarrow A \mid B \mid \dots \mid Z \mid a \mid b \mid \dots \mid z$

dígitos $\Rightarrow 0 \mid 1 \mid \dots \mid 9$

identificadores \Rightarrow letras (letras \mid dígitos)?

- a) **palavras reservadas:** a equipe deve descrever a linguagem denotada e escrever a definição regular de acordo com as palavras reservadas da linguagem 2021.1 usadas na QUESTÃO 1.

linguagem: _____

definição regular (padrão genérico): _____

- b) **identificadores:**

linguagem: $\{x \mid x \in (V_L \cup V_D \cup \{_ \})^+ \wedge x \text{ começa com uma letra ou com o símbolo } _, \text{ seguido ou não por letras, por dígitos ou pelo símbolo } _ \wedge x \text{ não pode terminar com dígito } \wedge x \text{ não pode ter dígitos ou } _ \text{ consecutivos }\}$.

definição regular: _____

- c) **constantes numéricas (inteiros e reais):**

linguagem: $\{x \mid x \in V_D^* \wedge |x| \geq 1 \wedge \text{se } x \text{ é uma constante inteira então possui no mínimo um e no máximo três dígitos } \wedge \text{se } x \text{ é uma constante real então possui no mínimo um e no máximo cinco dígitos, seguido(s) por ponto (casa decimal), seguido por no mínimo um e no máximo dois dígitos }\}$.

definição regular: _____

- d) **constantes literais:**

linguagem: $\{x \mid x \in \{\#0, \dots, \#255\}^* \wedge |x| \geq 2 \wedge x \text{ começa com aspas, seguida por zero ou mais caracteres da tabela ASCII, seguidos por aspas } \vee x \text{ começa com apóstrofo, seguido por zero ou mais caracteres da tabela ASCII, seguidos por apóstrofo }\}$.

definição regular: _____

- e) **comentário de linha:** a equipe deve descrever a linguagem denotada e escrever a definição regular.

linguagem: _____

definição regular: _____

- f) **comentário de bloco:** a equipe deve descrever a linguagem denotada e escrever a definição regular.

linguagem: _____

definição regular: _____

QUESTÃO 4. Construa **autômatos finitos determinísticos mínimos** que reconheçam as linguagens regulares denotadas pelas definições regulares correspondentes aos *tokens* do item anterior. Observe que as constantes inteiras e as constantes reais normalmente podem ser reconhecidas pelo mesmo autômato finito, visto que geralmente as regras de formação das primeiras são semelhantes as utilizadas para a formação das segundas. Essa observação também pode ser válida para os comentários de linha e de bloco ou para as palavras reservadas e identificadores (ou quaisquer outros *tokens*), conforme a definição feita. Se esta observação se aplicar à linguagem definida, então construa apenas um autômato para aqueles *tokens* para os quais isso é possível.

DATA LIMITE: trabalho a ser desenvolvido em aula

ENTREGAR: as especificações e os diagramas de transição dos AFDm

TRABALHO EM EQUIPE