AFD - Autômato Finito Determinístico

A solução aqui apresentada visa a implementaçãode um software capaz de simular a verificação de declaração de variáveis em C++.

✔ Conteúdos

🚀 Começando

O desenvolvimento desse Software é referente à Terceira Prova da disciplina de Linguagens Formais, Autômatos e Computabilidade da Universidade Federal do Pará - UFPA, que solicitava a implementação de um Autômato Finito Determinístico para processar Tipos e Nomes de Variáveis da linguagem C++.

Para fins de entendimento, há um arquivo com a extensão `.jff` referente a ideia do autômato que pode ser aberto com o software JFLAP. Além disso, há um arquivo jupyter (`.ipynb`) na pasta codeExplanation e um pdf com a explicação do código.

🛠️ Estrutura do AFD

Σ - Alfabeto de símbolos de Entrada;

```
import strings
I = {*(string.ascii_letters + "_")}
M = {*(string.ascii_letters + string.digits + "_")}
MDI = M.difference(I)
IUM = I.union(M)
```

Q – conjunto de estados possíveis do autômato;

```
{q0,q1,qf}
```

δ - Função de Transição ou Programa;

```
{"estado atual": {"símbolo processado": "estado alcançado"}}

transitions = {
    "q0":{
        "a": "q1", "b": "q1", ... ,"z": "q1", "A": "q1", "B": "q1", ...
,"Z": "q1", "_": "q1"
        },
    "q1":{
        "a": "q1", "b": "q1", ... ,"z": "q1", "A": "q1", "B": "q1", ...
,"Z": "q1", "_": "q1",
        "0":"q1", "1": "q1", ... ,"9":"q1", ",": "q0", ";":"qf"
        },
    "qf":{
        "a": None, "b": None, ... ,"z": None, "A": None, "B": None, ...
,"Z": None, "_": None,
        "0": None, "1": None, ... ,"9": None, ",": None, ";": None
        }
    }
```

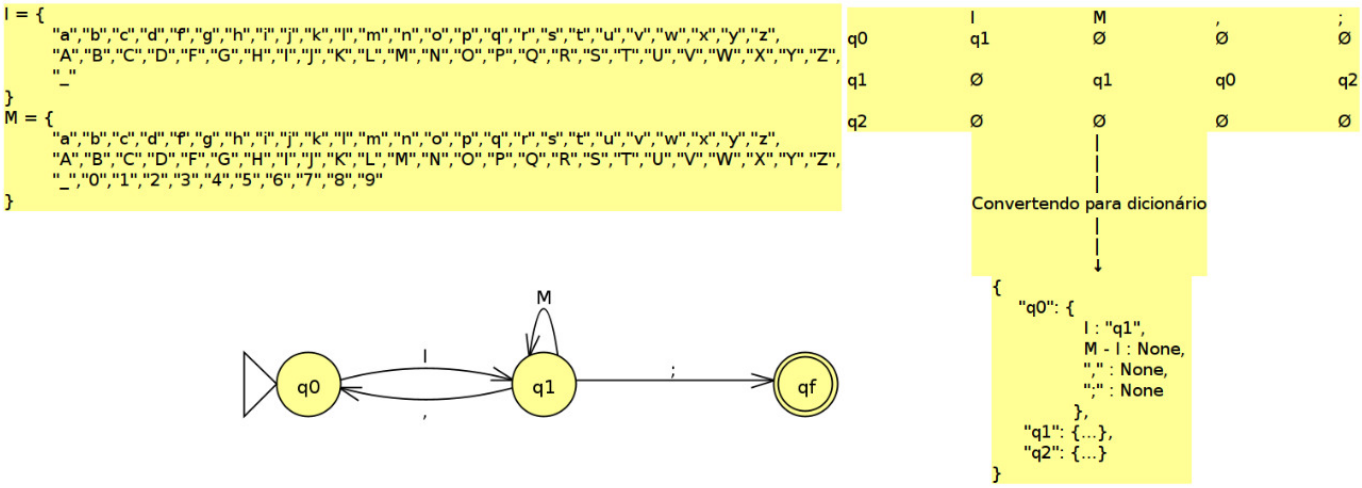
Estado inicial;

```
{q0}
```

Conjunto de estados finais;

```
{qf}
```

Representação.



🕹️ Funcionalidades

- Ler uma entrada em texto com as seguintes estruturas:
  - tipo_variavel nome_variavel;
  - tipo_variavel nome_variavel_1, nome_variavel_2, (...);
- **Processar a entrada** e **verificar se ela é aceita ou não**. Para isso, é verificado se o *tipo_variavel* é de algum tipo disponível na linguagem c++, que são :
  - char
  - int
  - bool
  - float
  - double
- Verifica se cada *nome_variave*l respeita as regras de nome das variáveis.
- Verifica se a linha de entrada termina corretamente com ";".

📦 Pré-requisitos

```
Python3+
```

💻 Como Executar

Abra o Terminal no diretório do software e digite o seguinte comando:

```
Python dfa.py
```

Entrada do usuário

- Para uma única variável

```
tipo_variavel nome_variavel;
```

- Mais de uma variável

```
tipo_variavel nome_variavel_1, nome_variavel_2, (...);
```

Possíveis Retornos

```
palavra aceita
```

```
palavra recusada
```

```
Erro de Declaração de Tipo
```

🎮 Exemplos

```
Python dfa.py
```

Palavra Aceita

```
int variavel1;
```

```
palavra aceita
Tipo primitivo: int
Variáveis: ['variavel1']
```

Palavra Recusada

```
bool 1Var;
```

```
palavra recusada
```

Erro de Declaração de tipo

```
chaars Var1;
```

```
Erro de Declaração de Tipo: chaars
```

- Para mais de uma declaração

```
bool var1, var1; char var 3
```

```
palavra aceita
Tipo primitivo: bool
Variáveis: ['var1', 'var1']
palavra recusada
```

👥 Equipe

👤**Aimeê Miranda Ribeiro** | **202104940014**

👤**Letícia Costa da Silva** | **202104940017**

👤**Luiz Jordany de Sousa Silva** | **202104940005**

👤**Syanne Karoline Moreira Tavares** | **202104920029**