Descrição do Código

Como Executar 🚀

```
$ python3 main.py [documento.txt] [entrada_aceita]
```

Exemplo:

```
$python3 main.py luc2.txt aab
```

O código que executa o autômato é divido em 4 classes: Estado, Transição, Finito e Contexto.

A classe Estado define informações de cada estado do autômato, salvando o nome do estado e uma lista com as possíveis transições a partir dele. Como pode se ver na Figura 1.

```
1  class Estado:
2      def __init__(self, nome):
3          self.nome = nome
4          self.transicao = []
5          pass
6
7      def addTransicao(self, transicao):
8          self.transicao.append(transicao)
9
10     def getNome(self):
11         return self.nome
12
```

Figura 1 - Classe Estado

A classe Finito como demonstra a Figura 2, representa o autômato como um todo, contendo o alfabeto de trabalho, o símbolo que representa o epsilon, uma lista com todos os estados deste autômato e uma lista com os estados de aceitação.

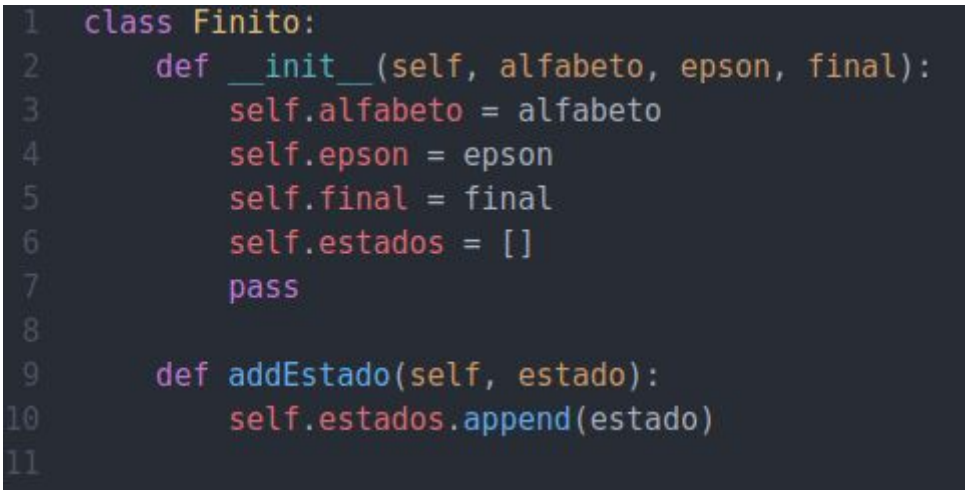

Figura 2 - Imagem representando a Classe Finito

A classe Transição descreve as possíveis transições realizada a partir de um determinado estado, ela contém o estado atual, símbolo que executa a transição, e o estado que é alcançado como resultado desta transição. Como apresenta a Figura 3.

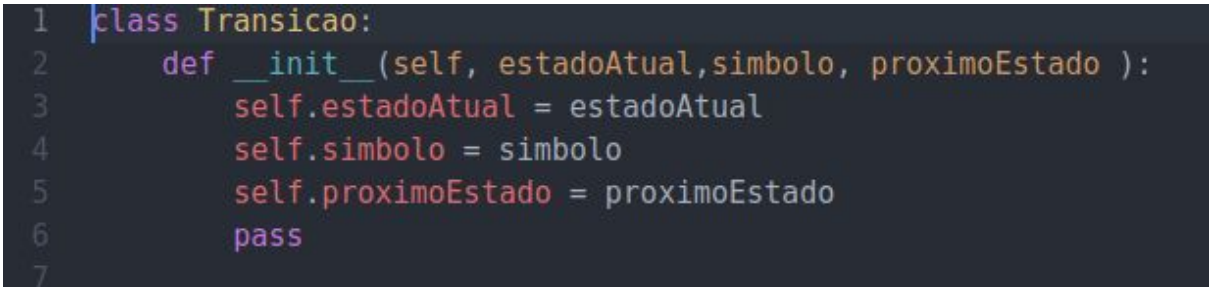

Figura 3 - Representação do Código da Classe Transicao

A classe Contexto é utilizada para autômatos finitos não determinísticos, esta classe contém uma cópia da palavra de entrada e estado atual. A Figura 4 representa bem esta classe.

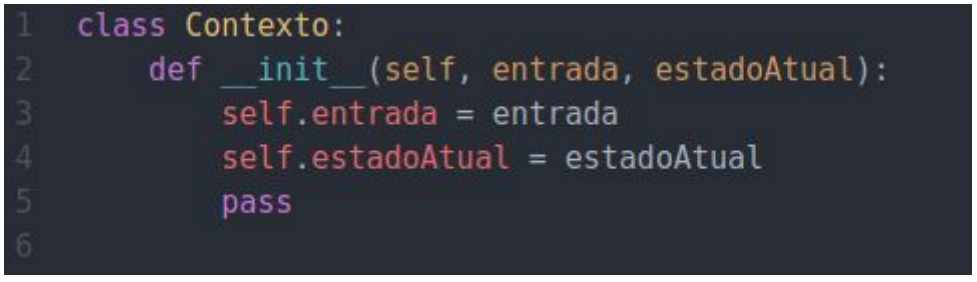

Figura 4 - Código da Classe Contexto

O arquivo main.py executa o automato e testa se a palavra de entrada é aceita ou rejeitada. Para isso é lido um arquivo txt contendo as informações para gerar este autômato, e também é lido uma palavra de entrada.

Cada informação lida no arquivo txt é salva em sua respectiva classe. Isto acontece dentro da função setup(), desde a linha 9 até a linha 50.O trecho a seguir de código cria uma lista

de contexto e define qual será o nosso primeiro contexto, no caso o "start" do autômato, logo após isso é chamada a função run().

Dentro da função run() é apresentado o estado inicial e a palavra de entrada. Em seguida é feito um laço de repetição, dentro dele é conferido se há transições a serem executadas ainda, trecho da linha 95 até 105. Dando continuidade, é feita uma checagem na transição verificando se esta é válida ou se é uma transição utilizando epsilon, a cada transição válida que não utiliza o símbolo epsilon é removida uma posição da palavra de entrada, no trecho da linha 92 até a 94 determina a retirada do contexto atual da lista de contexto acaso não encontrar nenhuma transição, assim indo para o próximo contexto. Logo após isso, é realizado um laço de repetição criando objetos da classe Contexto para realizar os diversos caminhos não determinísticos.