

Trabalho Final de Teoria da Computação

Nome: Miguel Rodrigues Guimarães de Oliveira

Matrícula: 201910339

Curso: GCC108 - Teoria da Computação

Professor: Rafael S. Durelli

Ideia Principal:

Desenvolver um programa em python que consiga ler um arquivo com a descrição de uma Máquina de Turing (seguindo uma determinada formatação padrão) e simular essa máquina com a entrada dada no arquivo, após a descrição da máquina. Ex:

```
(
    {q0, q1, q2, q3, q4, q5},
    {1},
    {1,B},
    {
        (q0, B) -> (q1, B, R),
        (q1, 1) -> (q1, 1, R),
        (q1, B) -> (q2, 1, R),
        (q2, 1) -> (q2, 1, R),
        (q2, B) -> (q3, B, L),
        (q3, 1) -> (q4, B, L),
        (q4, 1) -> (q5, B, L),
        (q5, 1) -> (q5, 1, L)
    }
    {q0}
)
B1111B111B
```

Estrutura de dados:

A estrutura de dados usada foi uma classe para representar um estado. Essa classe possui como atributos: um inteiro denominado valor e um vetor de dicionário, que salva as transições. Além disso, há três métodos: um para adicionar as transições, um para retornar algum dado de uma transição do estado, e outro para retornar a quantidade de transições do estado.

Funções:

limpa:

Esse método recebe duas strings: a primeira é a que será “limpa” e a segunda contém os caracteres que se deseja retirar da primeira.

cria_estados:

Esse método é responsável por ler o início do arquivo que contém a máquina, retornando ao final, um vetor que possui como tamanho, a quantidade de estados que a máquina representada possui.

transicoes:

É o método responsável por fazer a leitura das transições do arquivo, armazenando num vetor de estados, e retornando ao final de sua execução.

escreve_saida:

É o método responsável por escrever um determinado estado da máquina no arquivo de saída.

tem_transicao:

Esse método recebe um carácter que representa uma entrada na máquina, e um estado. Caso o estado em questão possua uma transição com aquele caractere de entrada, o método retorna a posição dessa transição no vetor de transições do estado. Caso não tenha, retorna -1.

Main:

A execução da main pode ser dividida em duas partes: entrada e saída. Na entrada, o arquivo de entrada é lido, e os estados importantes são armazenados: estados, alfabeto de entrada, alfabeto da fita, estado inicial e a entrada da fita. Para tratar e armazenar os dados do arquivo, são usados os métodos previamente citados (em especial o `limpa`).

Já na parte de saída, é aberto o arquivo de saída e a máquina de turing é de fato simulada. A simulação é feita em um laço, e a cada interação, um estado está “selecionado” uma linha é escrita no arquivo de saída. Além disso, a cada interação alguns dados são alterados, os quais são: a fita (é atribuída a saída dependendo do estado e da entrada da fita), a posição da fita (representando a cabeça de leitura, que vai para a direita ou esquerda) e o estado selecionado (dependendo também da entrada da fita e do estado que anteriormente era o selecionado).