

Maquina de Turing

Alberto Ferreira Neto

Introdução

- ◈ Trabalho desenvolvido usando HTML, CSS e JavaScript. Toda lógica do algoritmo que executa a Máquina de Turing e a manipulação dos elementos HTML foi feita com uso da linguagem de programação JavaScript. A máquina recebe um arquivo em JSON que representa a sêtucla, simula de forma gráfica a execução da MT que a sêtucla representa e finaliza informando se a MT aceita ou rejeita a cadeia.

Algoritmo

```
// Função para iniciar execução da Máquina de Turing
function executaMT(){

    const obj = JSON.parse(texto);
    var fita = palavra.split("");
    var estado = obj["EstadoInicial"];
    var pos = 0; // identifica posicao na fita
    var proxElemento = "";
    var direcao = 0; // Qual direcao vai a fita
    var msgFinal;
    var i = 0; // Contabilizador de tempo na execução das transições na tela
```

...

```
proxElemento = fita[pos];

// Retorna resultado da transicao
var transicao = transicaoSaida(obj, estado, proxElemento);
var entrada = transicao[0];
var saida = transicao[1];
estado = saida[0];
fita[pos] = saida[1];

if (transicao == "erro"){
    msgFinal = "Palavra não aceita";
    break;
}
console.log(entrada);
marcaCelula(pos, direcao, entrada, saida, i);

if (saida[2] == 'R'){
    direcao = 1;
    pos++;
} else{
    direcao = -1;
    pos--;
}

if (obj["EstadosFinais"].indexOf(estado) > -1){
    msgFinal = "Palavra aceita";
    break;
}
i++;
```

```
// Identifica a proxima transição da MT
function transicaoSaida(obj, estado, proxElemento){

    var result = [];

    for (var [key,value] of Object.entries(obj["Transicao"])){
        var entrada = key.split("-");
        var saida = value.split(",");

        if (entrada[0] == estado && entrada[1] == proxElemento){
            result.push(entrada);
            result.push(saida);

            return result;
        }
    }
    return "erro";
}
```


Sétupla

- Arquivo no formato JSON.
- Organizado por chave : valor.

```
1  {"Estados":["q0","q1","q2","q3","q4"],
2  "Alfabeto":["0","1"],
3  "AlfabetoFita":["0","1","X","Y","B"],
4  "Transicao":{"q0-0":"q1,X,R","q1-0":"q1,0,R","q2-0":"q2,0,L","q1-1":"q2,Y,L",
   "q1-X":"q2,X,R","q0-Y":"q3,Y,R","q1-Y":"q1,Y,R","q2-Y":"q2,Y,L","q3-Y":"q3,Y,R",
   "q0-B":"q4,B,R","q3-B":"q4,B,R","q2-X":"q0,X,R"}},
5  "EstadoInicial":"q0",
6  "SimboloBranco":"B",
7  "EstadosFinais":["q4"]}
```

MTs Testados

1. $L = \{0^n 1^n \mid n \geq 0\}$

Aceita

Palavra

0011

Iniciar Simulação

Fita

| | | | | |
|---|---|---|---|---|
| X | X | Y | Y | B |
|---|---|---|---|---|

Transição

q3,B -> q4,B,R

MTs Testados

1. $L = \{0^n 1^n \mid n \geq 0\}$

Rejeita

Palavra

00111

Iniciar Simulação

Fita

| | | | | | |
|---|---|---|---|---|---|
| X | X | Y | Y | 1 | B |
|---|---|---|---|---|---|

Transição

$q_3, Y \rightarrow q_3, Y, R$

MTs Testados

2. $\text{Max}(m-n, 0)$, $m-n$ se $m \geq n$ e 0 se $m < n$

Palavra

0000100

Iniciar Simulação

Fita

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| B | B | 0 | 0 | B | B | B | B |
|---|---|---|---|---|---|---|---|

Transição

$q_4, B \rightarrow q_6, 0, R$

Repositório no GitHub

- ◇ [alberttneto/SimuladorMT: Código feito com HTML, CSS e JAVASCRIPT, que simula uma máquina de turing; \(github.com\)](#)