

Universidade do Estado do Amazonas
Escola Superior de Tecnologia
Data: 15 de Novembro de 2019
Professora: Elloá B. Guedes
Disciplina: Fundamentos Teóricos da Computação
Monitor: Diego Lucena

PROJETO PRÁTICO III SIMULANDO UMA MÁQUINA DE TURING

1 Apresentação

A Tese de Church-Turing enuncia a equivalência entre máquinas de Turing e algoritmos. Assim, para reforçar seu entendimento destes conceitos, você deve criar um algoritmo que simula uma Máquina de Turing operando sobre uma entrada. Para exemplificar, suponha a máquina de Turing que aceita apenas números pares unários. Esta máquina de Turing é representada formalmente no diagrama de estados a seguir, construído com o auxílio do JFLAP.

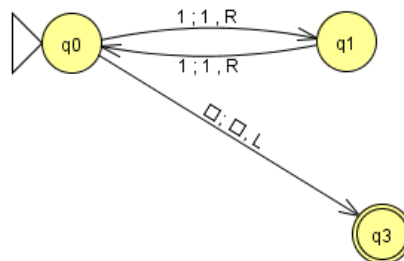


Figura 1: Diagrama de estados da máquina de Turing que aceita números pares unários.

Tomando esta máquina de Turing como base, é possível construir um programa em C que simula detalhadamente o funcionamento desta máquina. A fita é uma string e a posição do cabeçote é armazenada em uma variável do tipo inteira, que será utilizada para indexar a fita. Sempre que o cabeçote é movimentado, o estado da máquina é impresso. Os estados da máquina correspondem a funções na linguagem de programação. A programação em C desta máquina de Turing é mostrada a seguir.

```
1 #include <stdio.h>
2 #include <string.h>
3
4 char fita[80];
5 int cabecote = 0;
6 char *resultado = "REJEITA";
7
8
9 /*
10 Ler a entrada do usuario e descartar o \n ao final, inserindo \0
11 */
12 void inicializaFita(){
```

```
13     int ultimo;
14     fgets(fita,80,stdin);
15     ultimo = strlen(fita) - 1;
16     fita[ultimo] = '\\0';
17 }
18
19 void q0(){
20     if (fita[cabecote] =='1'){
21         cabecote++;
22         imprimeFita();
23         q1();
24         return;
25     } else if (fita[cabecote] == '\\0'){
26         cabecote--;
27         imprimeFita();
28         q3();
29         return;
30     }
31 }
32
33 void q1(){
34     if (fita[cabecote]== '1'){
35         cabecote++;
36         imprimeFita();
37         q0();
38         return;
39     } else {
40         return;
41     }
42 }
43
44 void q3(){
45     resultado = "ACEITA";
46     return;
47 }
48
49
50 void imprimeFita(){
51     int i = 0;
52     printf("Fita: ");
53     while (fita[i]!='\\0'){
54         if (i == cabecote){
55             putchar('>');
56         }
57         putchar(fita[i]);
58         i++;
59     }
60     printf("\\n");
61 }
62
63 int main(){
64     inicializaFita();
65     q0();
66     puts(resultado);
67     return 0;}
```

Ao executar o programa em questão para a entrada 111111 tem-se como resultado a tela de execução mostrada na figura a seguir.

```

111111
Fita: [1>11111]
Fita: [11>1111]
Fita: [111>111]
Fita: [1111>11]
Fita: [11111>1]
Fita: [111111]
Fita: [111111>1]
ACEITA

-----
Process exited after 6.366 seconds with return value 0
Pressione qualquer tecla para continuar. . . _

```

2 Especificação do Projeto

Você deve construir uma máquina de Turing capaz de decidir a seguinte linguagem, em que $\Sigma = \{ , a, b, c \}$:

$$L = \{ w\# | w \in \Sigma^* \wedge \#a's < \#b's < \#c's \wedge \#a's \geq 1 \},$$

ou seja, aceitar a linguagem das palavras compostas por a's, b's e c's e espaços em branco, sempre terminadas por # cuja quantidade de a's é menor que a de b's que, por sua vez, é menor que a de c's. Alguns exemplos de palavras pertencentes à esta linguagem são: abbccc#, aa bbb ccccc#, cccb b a#, a b b b c c c c c #, etc. Todas as palavras que não pertencem à esta linguagem devem ser rejeitadas. Note que a cardinalidade desta linguagem é infinita e que a mesma é decidível.

Primeiramente, conceba uma máquina de Turing capaz de resolver este problema com o auxílio do JFLAP. Nesta ferramenta, efetue testes que auxiliem a avaliar se a solução proposta resolve o problema apresentado. Uma vez concluída esta etapa, o passo seguinte consiste em converter esta máquina de Turing para a linguagem de programação Python, considerando uma string como entrada e produzindo uma string na saída, com a impressão da fita original e a palavra “ACEITA” ou “REJEITA”, indicando o estado final da máquina. Note que a entrada não deve ser modificada ao computar a palavra.

Considerando a fidelidade ao simular máquinas de Turing, não faça uso de tipos numéricos na sua linguagem de programação nem tampouco de operadores relacionais para checar as quantidades. Aqueles que procederem diferente desta especificação terão pontuação anulada. Você pode construir sua máquina de Turing utilizando qualquer artifício das variantes mostradas em sala de aula.

3 Exemplos

Entrada	Saída
aabbcc#	aabbcc# REJEITA
a a b b b c c c c #	a a b b b c c c c # ACEITA
bbcaaa#	bbcaaa# REJEITA
bbccc#	bbccc# REJEITA

4 Datas Importantes

- Abertura para submissões: 14/11/2019, 16h30min (horário do servidor)
- Entrega: 26/11/2019, 23h55min (horário do servidor)