

Fundamentos Teóricos da Computação

Atividade Complementar

Elloá B. Guedes

Escola Superior de Tecnologia
Universidade do Estado do Amazonas
Av. Darcy Vargas, 1200 – Manaus, AM
ebgcosta@uea.edu.br

1 de julho de 2024

1. Apresentação

A Máquina de Turing é uma máquina de computação abstrata com o poder dos computadores reais e de outras definições matemáticas do que pode ser calculado. A Máquina de Turing (MT) consiste em um controle de estados finitos e em uma fita infinita dividida em células. Cada célula contém um único símbolo, dentre um conjunto finito de símbolos da fita, e uma célula está na posição atual do cabeçote da fita. A MT executa movimentos com base em seu estado atual e no símbolo de fita presente na célula sob o cabeçote. Em um movimento, ela muda de estado, sobregrava a célula varrida com algum símbolo de fita e move o cabeçote uma célula para a esquerda ou direita.

É possível, em princípio, simular uma MT por um computador real, se aceitarmos que existe um suprimento potencialmente infinito de um dispositivo de armazenamento removível como um disco, a fim de simular a parte não branca da fita da MT. Tendo em vista que os recursos físicos para produzir discos não são infinitos, esse argumento é questionável. Porém, como os limites sobre a quantidade de espaço de armazenamento existente no universo são desconhecidos e sem dúvida vastos, a hipótese de um recurso infinito, como na fita da MT, é realista na prática e geralmente aceita.

Uma MT, por sua vez, pode simular o espaço de armazenamento e o controle de um computador real, usando uma única fita para armazenar todos os locais e seu conteúdo: registradores, memória principal, discos e outros dispositivos de armazenamento. Desse modo, podemos ter confiança em que algo que não possa ser feito por uma MT não poderá ser realizado por um computador real.

A tese de Church-Turing pode ser assim enunciada:

“Se uma função é efetivamente computável, então ela é computável por meio de uma máquina de Turing.”

Ou, equivalentemente:

“Todo algoritmo pode ser expresso mediante uma máquina de Turing.”

Diante do exposto, este projeto prático consiste em reforçar o seu entendimento desta tese por meio de uma simulação. O seu objetivo, portanto, consiste em simular uma máquina de Turing que resolve um problema utilizando uma linguagem de programação.

Para exemplificar, suponha a máquina de Turing que decide números pares unários. Esta máquina de Turing é representada no diagrama de estados a seguir, construído com o auxílio do JFLAP.

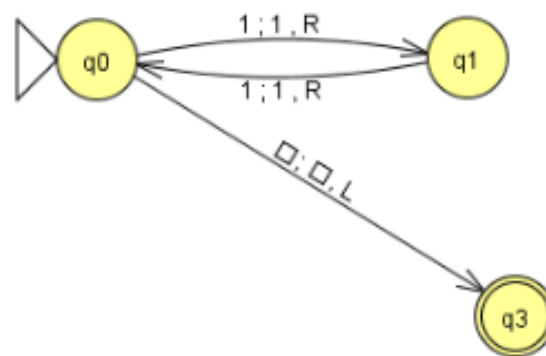


Figura 1: Diagrama de estados da máquina de Turing que decide números pares unários.

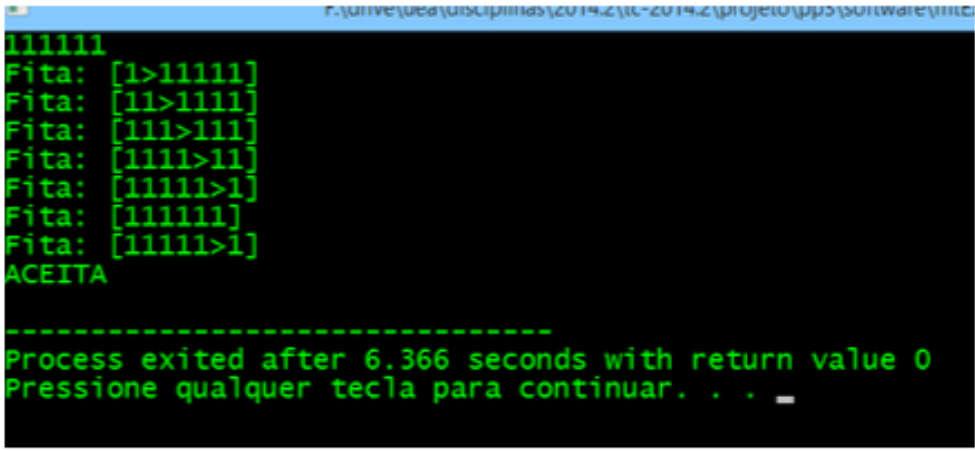
Tomando esta máquina de Turing como base, é possível construir um programa em C que simula detalhadamente o funcionamento desta máquina. A fita é uma string e a posição do cabeçote é armazenada em uma variável do tipo inteira, que será utilizada para indexar a fita. Sempre que o cabeçote é movimentado, o estado da máquina é impresso. Os estados da máquina correspondem às 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 void q0();
8 void q1();
9 void q3();
10 void imprimeFita();
11 /*
12 Ler a entrada do usuario e descartar o \n ao final, inserindo \0
13 */
14 void inicializaFita(){
15     int ultimo;
```

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

```
65
66 int main() {
67     inicializaFita();
68     q0();
69     puts(resultado);
70 }
```

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



2. Especificação do Projeto

O projeto deve ser desenvolvido em duplas utilizando a linguagem de Programação Python 3.4 ou superior. Não devem ser utilizadas bibliotecas externas. A atividade vale, no máximo, 30% dos pontos da AP3.

As duplas deve construir uma máquina de Turing decidora capaz de reconhecer palavras pertencentes à linguagem $x\#y$, em que x e y são números denotados na linguagem unária com símbolos I . A máquina, além de reconhecer as palavras da linguagem especificada, deve imprimir, ao final da entrada, o resultado de $x \bmod y$ e a palavra “ACEITA”. Quando não for possível, deve escrever apenas a palavra “REJEITA”.

Comece a resolver o projeto construindo esta máquina de Turing no JFLAP. O passo seguinte é 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 saída na fita e mais a palavra “ACEITA” ou “REJEITA”, indicando o estado final da máquina.

Considerando a fidelidade ao simular máquinas de Turing, não faça uso de tipos numéricos na sua linguagem de programação. Aqueles que procederem diferente desta especificação terão pontuação cortada pela metade. Lembre-se que o objetivo é fazer uma máquina de Turing para o problema em questão.

Você pode construir sua máquina de Turing utilizando qualquer artifício das variantes mostradas em sala de aula: multi-fitas, enumeradora, cabeçote que pára, não determinística, etc. Para fins de simplificação, não serão consideradas saídas nulas, nem entradas ou resultados negativos.

3. Exemplos

Entrada	Saída
IIIIII#III	IIIIII#III=I ACEITA
I#I#I#	I#I#I# REJEITA
IIIIII#IIII	IIIIII#IIII=III ACEITA
IIIIIII#IIII	IIIIIII#IIII=IIII ACEITA
IIIIIIIIIIII	IIIIIIIIIIII REJEITA

4. Links Úteis

- <<https://www.youtube.com/watch?v=FTSAiF9AHN4>>
- <https://en.wikipedia.org/wiki/History_of_the_Church%E2%80%93Turing_thesis>