

Universidade do Estado do Amazonas
Escola Superior de Tecnologia
Data: 10 de Maio de 2018
Professora: Elloá B. Guedes
Disciplina: Fundamentos Teóricos da Computação

PROJETO PRÁTICO III SIMULANDO UMA MÁQUINA DE TURING COM PYTHON

1 Apresentação

A Tese de Church-Turing enuncia a equivalência entre máquinas de Turing e algoritmos. Assim, pode-se dizer que os programas que você constrói na sua linguagem de programação favorita são máquinas de Turing!

Este projeto prático consiste em reforçar o seu entendimento da Tese de Church-Turing por meio de uma simulação. O objetivo é simular uma máquina de Turing que resolve um problema utilizando uma linguagem de programação.

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

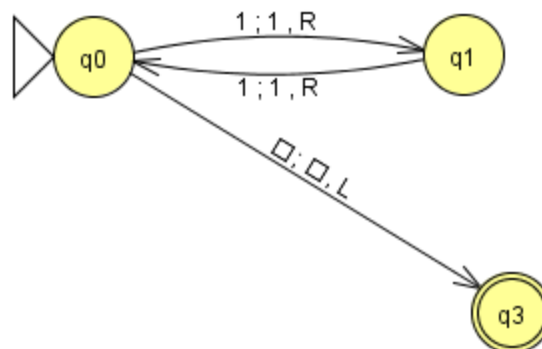


Figura 1: Diagrama de estados da máquina de Turing que aceita números pares unários.

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.

```
#include <stdio.h>
#include <string.h>

char fita[80];
int cabecote = 0;
char *resultado = "REJEITA";
void q0();
void q1();
void q3();
void imprimeFita();
/*
Ler a entrada do usuario e descartar o \n ao final, inserindo
\0
*/
void inicializaFita(){
    int ultimo;
    fgets(fita,80,stdin);
    ultimo = strlen(fita) - 1;
    fita[ultimo] = '\0';
}

void q0(){
    if (fita[cabecote] == '1'){
        cabecote++;
        imprimeFita();
        q1();
        return;
    } else if (fita[cabecote] == '\0'){
        cabecote--;
        imprimeFita();
        q3();
        return;
    }
}

void q1(){
    if (fita[cabecote] == '1'){
```

```
        cabecote++;
        imprimeFita();
        q0();
        return;
    } else {
        return;
    }
}

void q3(){
    resultado = "ACEITA";
    return;
}

void imprimeFita(){
    int i = 0;
    printf("Fita: [");
    while (fita[i] != '\0'){
        if (i == cabecote){
            putchar('>');
        }
        putchar(fita[i]);
        i++;
    }
    printf("]\n");
}

int main(){

    inicializaFita();
    q0();
    puts(resultado);
}
```

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: [11111>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 realizar a soma de números em binário. A entrada da sua máquina de Turing são os dois números separados pelo o operador de soma. A sua máquina de Turing para este problema deve escrever na fita, ao final da entrada, o resultado da operação, também na notação binária. 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 fita e 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 para armazenar os operandos e nem use o operador de soma da linguagem de programação. Aqueles que procederem diferente desta especificação terão pontuação anulada. Lembre-se que o objetivo é fazer uma máquina de Turing para a operação aritmética considerada.

Você pode construir sua máquina de Turing utilizando qualquer artifício das variantes mostradas em sala de aula.

3 Exemplos

Entrada	Saída
001+001	001+001=010 ACEITA
010-001	010-001 REJEITA
1010101010110111+01	1010101010110111+01=1010101010111000 ACEITA
001	001 REJEITA

4 Links Úteis

- <http://www.csgnetwork.com/binaddsubcalc.html>
- <https://www.youtube.com/watch?v=S9LJknZTyos>
- <http://producao.virtual.ufpb.br/books/camyle/introducao-a-computacao-livro/livro/livro.chunked/ch03s05.html>

Mãos à obra!
“Keep calm and carry on.”