

Nome: Wyllen Brito da Silva
Disciplina: Teoria da Computação (TEC)
Udesc Joinville
Profª Karina Roggia

Introdução – Descrição do Trabalho

O objetivo deste trabalho é trabalhar a simulação entre modelos de máquina de Turing. Para isto, será utilizada a sintaxe disponível no simulador online de máquinas de Turing em <http://morphett.info/turing/turing.html>

O trabalho consistirá na programação de um "tradutor" de modelos de máquina de Turing. A entrada será um arquivo texto com extensão .in com um programa para o simulador online, consistindo em um programa de máquina de Turing de acordo com a sintaxe do simulador. A primeira linha do arquivo será

- ;S caso o programa de entrada seja um programa para o modelo de Sipser (fita com início à esquerda), ou
- ;I caso o programa de entrada seja um programa para o modelo de fita duplamente infinita.

A saída deve ser um arquivo texto com extensão .out com um programa para o modelo "oposto" ao de entrada, capaz de ser executado no simulador (podendo fazer uso de movimento estacionário).

Os arquivos serão constituídos da primeira linha de comentário conforme explicado anteriormente seguidas exclusivamente de linhas no formato

<current state> <current symbol> <new symbol> <direction> <new state>

conforme especificado no site do simulador. O estado inicial será sempre nomeado como 0. O alfabeto da fita será constituído exclusivamente de letras (maiúsculas e minúsculas) e dígitos, portanto símbolos como #, &, %, £, ¢, § não serão utilizados em nenhum arquivo de entrada, podendo ser utilizados como eventuais símbolos auxiliares.

Os arquivos de entrada serão apenas de máquinas de Turing determinísticas e com codificação válida para o simulador. Todos os programas dados como entrada serão para reconhecimento de linguagens sobre o alfabeto {0,1}. Note, portanto, que a máquina de Turing dada como saída também será um reconhecedor para essa mesma linguagem sobre o alfabeto {0,1}.

O programa deve ser possível de ser executado em um computador com sistema operacional Ubuntu 22.04.4 LTS de 64 bits. A linguagem de implementação é livre dentro destas restrições.

A entrega deverá conter:

1. o programa final,
2. instruções claras de execução (os testes de avaliação serão realizados a partir de tais instruções), e
3. endereço de repositório público no github com todo o código-fonte.

Casos de teste:

- O arquivo odd.txt é de uma máquina com fita limitada à esquerda (modelo de Sipser) que aceita a linguagem das sequências binárias de comprimento ímpar.
- O arquivo sameamount10.txt é de uma máquina com fita duplamente infinita que aceita a linguagem das sequências binárias que possuem a mesma quantidade de 0s e de 1s.

Construção das máquinas de turing (MT)

A ideia é construir duas máquinas de turing que fazem a conversão de uma máquina para outra. Foi criado um programa em python que lê um arquivo texto de entrada com extensão .in e captura o conteúdo da primeira linha do arquivo, se for ;S o programa entende que é para simular uma MT de sipser em uma MT com fita duplamente infinita, e se for ;I o programa entende que é para fazer o oposto.

Instruções de execução

Substitua o caminho do arquivo:

```
if __name__ == "__main__":  
  
    path_MT_Sipser = "Casos_de_teste\\odd.in"  
    path_MT_Duplamente_Infinita = "Casos_de_teste\\sameamount10.in"  
  
    path = path_MT_Duplamente_Infinita
```

O próprio programa se encarregará de fazer a leitura da primeira linha do arquivo e definir se é ;S ou ;I e fazer a conversão necessária:

```
if primeira_linha == ";S":  
    print("\nO arquivo lido é uma MT de Sipser\n")  
    print("**** Realizando conversão para simular essa MT Sipser em uma MT Duplamente Infinita  
    ****")  
    Simula_Sipser_em_MTD(path)  
    print("\nSUCESSO! VERIFIQUE O ARQUIVO: \"saida.out\"\n")  
    print("No simulador Use STANDARD e q0 como símbolo inicial")  
elif primeira_linha == ";I":  
    print("\nO arquivo lido é uma MT Duplamente Infinita\n")  
    print("**** Realizando conversão para simular essa MT Duplamente Infinita em uma MT Sipser  
    ****")  
    Simula_MTD_em_Sipser(path)  
    print("\nSUCESSO! VERIFIQUE O ARQUIVO saida.out\n")  
    print("No simulador Use Semi-Infinite Tape e q0 como símbolo inicial")
```

No simulador, sempre utilize q0 como símbolo inicial.

No simulador, para a máquina de turing com primeira linha “;S” significa que ela é uma máquina de turing de Sipser, logo deve ser simulada no simulador com a fita duplamente inifita, nesse caso, a “Standard”.

The screenshot shows the 'Controls' panel of a Turing Machine simulator. It includes buttons for 'Run', 'Pause', 'Step', and 'Reset'. A checkbox labeled 'Run at full speed' is checked. An 'Undo' button is located to the right of the 'Step' button. Below these buttons is a text input field for 'Initial input' containing the string '1001111'. A link for 'Advanced options' is present. Below that is a text input field for 'Initial state' containing 'q0'. A dropdown menu for 'Machine variant' is set to 'Standard'. Below the dropdown is a descriptive text: 'Standard Turing machine with tape infinite in both directions'. At the bottom are two links: 'Load an example program' and 'Save to the cloud'.

Controls

Run ☒ Run at full speed

Pause

Step Undo

Reset

Initial input: 1001111

[Advanced options](#)

Initial state: q0

Machine variant: Standard ▼

Standard Turing machine with tape infinite in both directions

[\[Close\]](#)

[Load an example program](#)

[Save to the cloud](#)

No simulador, para a máquina de turing com primeira linha “;I” significa que ela é uma máquina de turing com fita duplamente infinita, logo deve ser simulada no simulador com a fita limitada à esquerda, nesse caso, a “Semi-Infinite Tape”.

The screenshot shows the 'Controls' panel of a Turing Machine simulator. It includes buttons for 'Run', 'Pause', 'Step', and 'Reset'. A checkbox labeled 'Run at full speed' is checked. An 'Undo' button is located to the right of the 'Step' button. Below these buttons is a text input field for 'Initial input' containing the string '001101'. A link for 'Advanced options' is present. Below that is a text input field for 'Initial state' containing 'q0'. A dropdown menu for 'Machine variant' is set to 'Semi-infinite tape'. Below the dropdown is a descriptive text: 'Turing machine with tape infinite in one direction only (as used in, eg, Sipser)'. At the bottom are two links: 'Load an example program' and 'Save to the cloud'.

Controls

Run ☒ Run at full speed

Pause

Step Undo

Reset

Initial input: 001101

[Advanced options](#)

Initial state: q0

Machine variant: Semi-infinite tape ▼

Turing machine with tape infinite in one direction only (as used in, eg, Sipser)

[\[Close\]](#)

[Load an example program](#)

[Save to the cloud](#)

Simulando uma MT de Sipser em uma MT com fita duplamente Infinita

Em uma fita duplamente infinita, utilizar o símbolo auxiliar “#” (que faz parte do alfabeto da fita) antes de qualquer processamento da linguagem, a fim de delimitar o começo da fita (como a fita é infinita, e queremos simular uma MT de Sipser que é infinito à direita, utilizaremos o símbolo “#” para delimitar o começo dessa fita).



Figura 1: O símbolo “#” delimita o início da fita em uma MT duplamente infinita.

Se em algum momento do processamento da máquina de turing, chegarmos à esse símbolo “#”, posicionaremos o cabeçote da fita no símbolo imediatamente à direita desse símbolo “#”.

Representação da inserção do símbolo “#” antes de qualquer processamento da máquina:

$q_0 * * L q_1$

$q_1 * \# R 0$

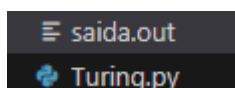
Onde o símbolo asterisco “*” representa o símbolo coringa em <current symbol> ou <current state> para corresponder a qualquer caractere ou estado.

Como pela definição do trabalho, o estado inicial sempre será 0, será possível com que ocorra a representação da maneira como foi mostrada anteriormente.

É importante ressaltar que para esse programa que faz a bisssimulação entre os modelos de máquinas de turing, foi adotado o estado q0 como estado inicial.

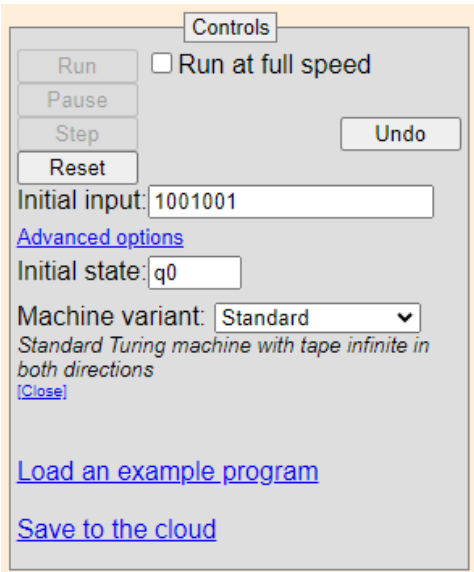
Rodando o programa, para simular uma MT Sipser em MT com fita duplamente infinita:

O programa Turing.py produz o arquivo texto de saída: *saida.out*



Rodaremos o conteúdo do arquivo saída.out, que recebe o arquivo texto ood.in, que é do tipo máquina de turing de sipser, dentro do simulador fornecido.

Usaremos o tipo de fita Standard, que é a fita infinita para ambos os lados e modificaremos o estado inicial para q0, conforme imagem abaixo:



Assim, copiaremos o conteúdo do arquivo saída.out e colaremos no espaço reservado para a máquina de turing no simulador, um exemplo de saída para a entrada 111 é aceito pela máquina gerada pelo simulador:



Simulando uma MT com fita duplamente infinita em uma MT de Sipser

Antes de iniciar qualquer processamento na fita com limitação à esquerda, deve-se matar o símbolo inicial e final, para que se delimite o começo e final da fita.

Símbolo delimitar Inicial da fita : #

Símbolo delimitar Final da fita: \$

Sempre que for necessário colocar um símbolo mais à esquerda da fita, empurraremos todo o conteúdo da fita para a direita, o quanto for necessário.

O que passa a delimitar o final da fita agora não é mais o símbolo branco após um símbolo, mas sim o símbolo que representa o final da fita \$.

Se em algum momento, realizando o movimento à esquerda, for encontrado o símbolo #, devo empurrar todo o conteúdo da fita à direita e inserir o símbolo branco _ imediatamente à direita do símbolo # e posicionar o meu cabeçote da fita nesse símbolo branco _.

Se em algum momento, realizando o movimento à direita eu encontrar o símbolo \$, que delimita o final da fita, eu empurrarei esse símbolo \$ para a direita e escreverei o símbolo branco _ no lugar dele, e retornarei o cabeçote para esse mesmo símbolo branco.

Representação da inserção do símbolo “#” e “\$” como delimitadores da fita antes de qualquer processamento pela máquina original:

q0 1 % R q1

q0 0 & R q1

q1 * * R q1

q1 _ _ R q2

q2 _ \$ L q3

q3 _ _ L q3

q3 1 _ R q4

q4 _ 1 L q3

q3 0 _ R q5

q5 _ 0 L q3

q3 % # R q6

q3 & # R q7

q6 _ 1 L q8

q7 _ 0 L q8

q8 # * R 0

Além disso, para cada estado, será necessário fazer uma máquina de turing que processe a leitura de # e \$ no começo e final da fita, a fim de que o estado seja guardado e também o cabeçote seja posicionado no lugar necessário:

Apenas um exemplo, para o estado zero, a seguinte rotina é executada, quando se encontra o valor # no começo da fita, todo o conteúdo é empurrado para direita e posiciona-se o cabeçote no símbolo imediatamente à direita de #.

```
0 # * R q30
q30 * * R q30
q30 $ _ R q31
q31 _ $ L q32
q32 _ _ L q32
q32 1 _ R q33
q33 _ 1 L q32
q32 0 _ R q34
q34 _ 0 L q32
q32 # * R 0
```

A rotina abaixo é executada, quando encontra-se o símbolo \$ no final da fita, o símbolo \$ é empurrado para a direita, e abre-se um espaço para realizar o processamento necessário no final da fita:

```
0 $ _ R q8
q8 _ $ L q9
q9 _ * * 0
```

Um exemplo da máquina rodando com o código obtido, sob a entrada 01011010:

The screenshot displays a Turing Machine (TM) simulator interface. At the top, the 'Tape' is shown with the content '# BXXXXXXXXX \$'. Below the tape, the 'Current state' is 'halt-accept' and the machine is 'Halted.'. The 'Steps' counter shows 173. The 'Turing machine program' section lists the following code:

```

215 9 # * R q77
216 q77 * * R q77
217 q77 $ _ R q78
218 q78 _ $ L q79
219 q79 _ _ L q79
220 q79 1 _ R q80
221 q80 _ 1 L q79
222 q79 0 _ R q81
223 q81 _ 0 L q79
224 q79 # * R 9
225
226
227 10 # * R q82
228 q82 * * R q82
229 q82 $ _ R q83
230 q83 _ $ L q84
231 q84 _ _ L q84
232 q84 1 _ R q85
233 q85 _ 1 L q84
234 q84 0 _ R q86
235 q86 _ 0 L q84
236 q84 # * R 10
237

```

The 'Controls' panel on the right includes buttons for 'Run', 'Pause', 'Stop', and 'Reset'. The 'Run' button is checked, and the 'Run at full speed' checkbox is also checked. The 'Initial input' is '01011010', and the 'Initial state' is 'q0'. The 'Machine variant' is set to 'Semi-infinite tape'. There are links for 'Advanced options', 'Load an example program', and 'Save to the cloud'.