



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”

FCT – Faculdade de Ciências e Tecnologia

DMC – Departamento de Matemática e Computação

Bacharelado em Ciência da Computação

Simulador de Máquinas de Turing - v1

RELATÓRIO

Disciplina: Teoria da Computação

Docente: Celso Olivete Júnior

Discentes: Carolina Dias Chaves

João Vítor Antunes Ribeiro

Presidente Prudente, 6 de Outubro de 2014

Sumário

1. Objetivos.....	3
2. Descrição.....	3
3. Simulador de Máquinas de Turing Multifitas.....	3
3.1. Criação/Edição/Remoção de MTMs.....	3
3.2. Criação/Edição/Remoção de estados ou transições.....	4
3.3. Reconhecimento de uma ou mais entradas.....	6
3.4. Reconhecimento de uma entrada por meio de simulação passo a passo.....	7
3.5. Abrir/Salvar uma MTM no mesmo formato que o JFlap (.jff).....	8
4. Conclusões.....	8

1. Objetivos

Esse trabalho tem como objetivo desenvolver o entendimento de Linguagens Formais e seu potencial de representação através da implementação de um simulador de linguagens de autômatos formais. Durante a disciplina de Teoria da Computação, foram desenvolvidos dois simuladores. No primeiro, o principal objetivo foi a implementação de um Simulador Universal de Máquinas de Turing. Já nessa segunda parte, o primeiro simulador foi estendido, tendo sido implementado um Simulador de Máquinas de Turing Multifitas (MTMs).

2. Descrição

O desenvolvimento da aplicação foi feito na linguagem Java, utilizando a plataforma Eclipse. Os requisitos implementados podem ser descritos como a especificação e simulação de Máquinas de Turing Multifitas através de diagramas de transições (autômatos) que possam permitir a/o:

1. Criação/Edição/Remoção de MTMs;
2. Criação/Edição/Remoção de estados ou transições;
3. Reconhecimento de uma ou mais entradas;
4. Reconhecimento de uma entrada por meio de simulação passo a passo.
5. Abrir/Salvar uma MTM no mesmo formato que o JFlap (.jff).

3. Simulador de Máquinas de Turing Multifitas

3.1. Criação/Edição/Remoção de MTMs

Na Figura 1 (em cima), podemos observar os botões para criação (verde) e remoção (vermelho) de máquinas. Ao clicar sobre o botão de criação, uma aba é criada para permitir a criação de uma nova MTM, sem antes exibir uma janela de seleção do número de fitas que deverá ser usado, como mostra a Figura 1 (embaixo).

Por definição, a Máquina de Turing Multifitas deve permitir a criação de autômatos com número de fitas variando de 1 à N , $N \geq 2$, mas por limitações de linguagem, o máximo número inteiro permitido em Java, para administrar o processo de simulação, é 2.147.483.647. Portanto, o simulador desenvolvido suporta MTM com fitas variando de 1 à 2.147.483.647.

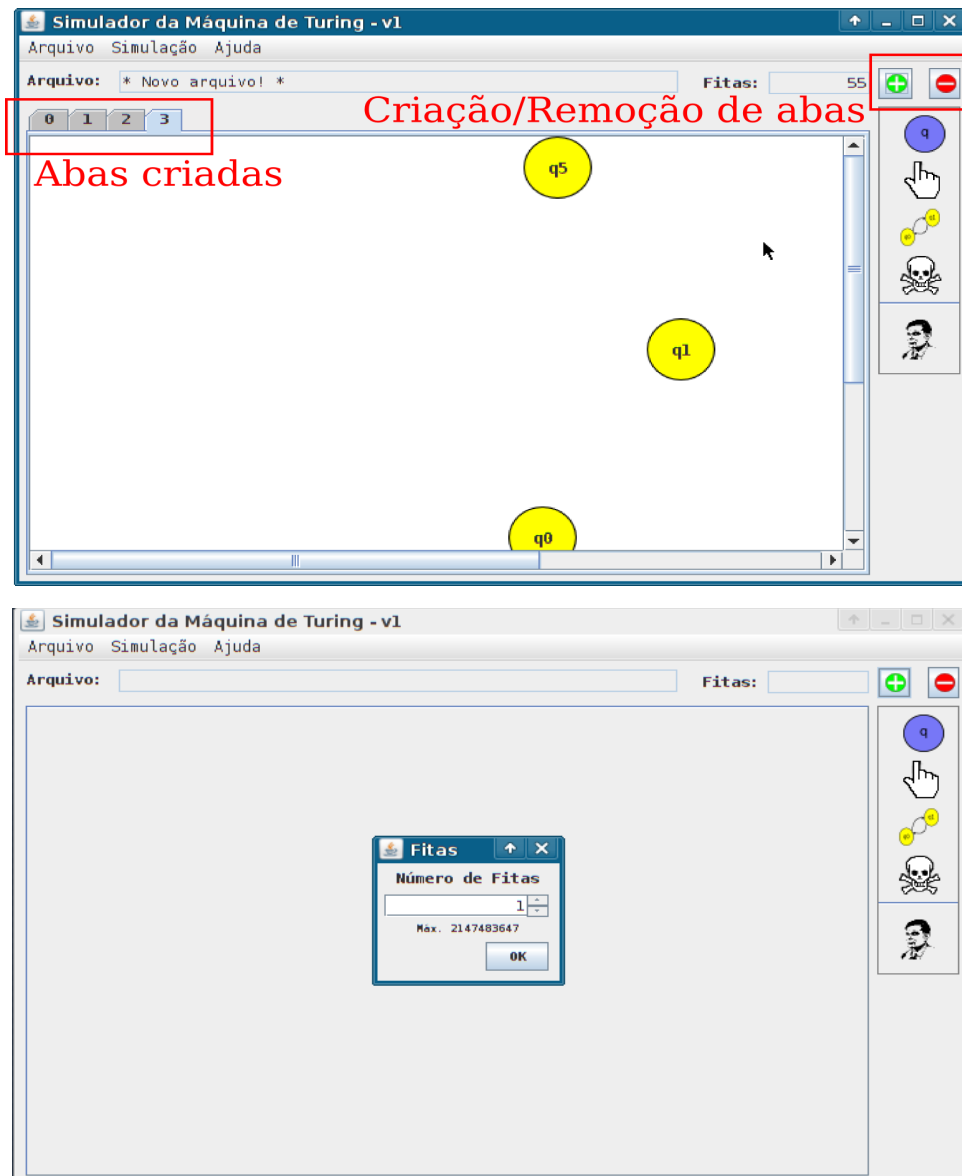


Figura 1: Criação de múltiplas MTFs: abas criadas (em cima) e definição do número de fitas (embaixo).

3.2. Criação/Edição/Remoção de estados ou transições

Na Figura 2, podemos observar o conjunto de botões para criação de máquinas por meio de autômatos; ao clicar sobre algum desses botões e depois sobre o painel de criação (com redimensionamento automático quando um estado é movido fora da área atualmente visível), é possível criar/editar/remover/mover estados ou transições. O botão de movimento, representado por uma mão, permite que, ao selecioná-lo, seja possível arrastar os estados pelo painel (região em branco). Em seguida, o botão de transição permite que, ao selecioná-lo, seja possível criar uma transição entre dois estados, clicando com o botão esquerdo do mouse em um estado e depois sobre o outro estado que se deseja finalizar a transição; dessa forma, será criada uma transição entre esses estados. Por fim, o botão de remoção permite a remoção de transição ou estados.

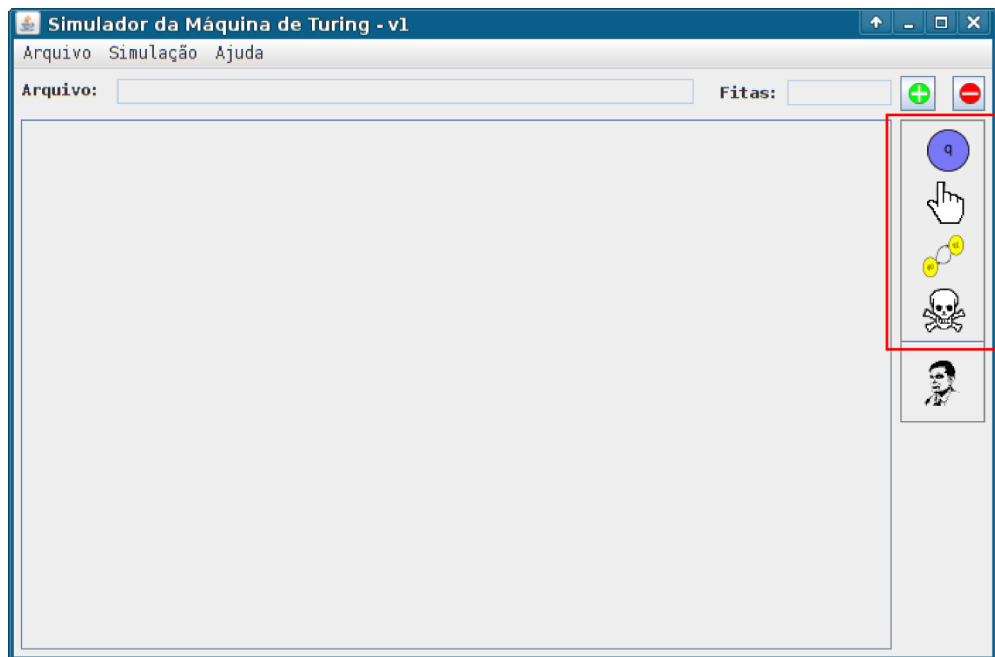


Figura 2: Funcionalidades do autômato.

Para criar uma transição, deve-se clicar em dois estados, de onde deve sair a transição e onde a mesma deve entrar. Após clicar no segundo estado (que pode ser o mesmo), uma janela (ilustrada na Figura 3) será aberta para que o usuário possa introduzir os dados da transição, isto é, os símbolos que serão lidos, os que serão escritos e para quais lados as fitas vão se locomover quando finalizar a transição (ou direita – D ou esquerda – E ou parado – P). Para editar uma transição é necessário dar dois cliques (botão esquerdo) sobre o símbolo da transição, estando a funcionalidade de movimento ativada. Os caracteres aceitos são todos os 10 dígitos, todas as 26 letras maiúsculas e minúsculas (26, sem acento ou ç), □ e espaço (ambos representando o vazio), ressaltando que apenas um único símbolo é aceito em cada campo de leitura ou escrita.

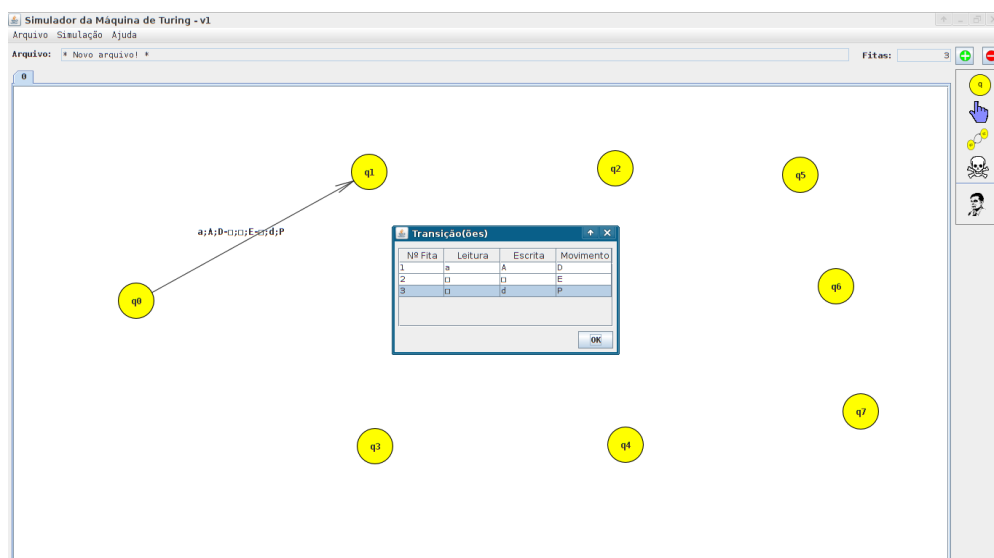


Figura 3: Transição entre estados.

A troca de estados (inicial/final/normal) pode ser realizada ao clicar com o botão direito do mouse sobre o estado que se deseja alterar, com a funcionalidade de movimento ativa, e em seguida escolher o tipo de estado para

o qual deseja-se transformá-lo e clicar em OK, conforme ilustrado na Figura 4.

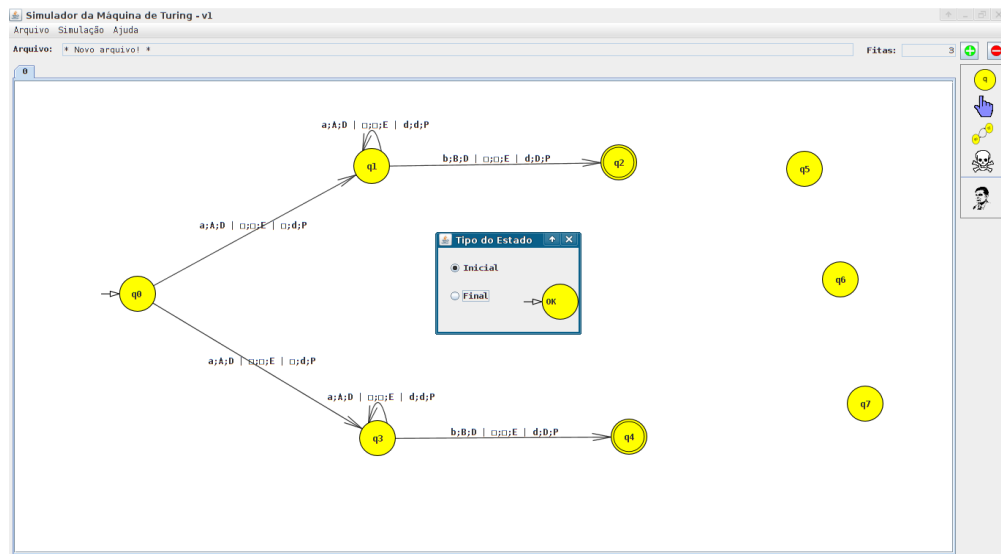


Figura 4: Mudança de estados.

3.3. Reconhecimento de uma ou mais entradas

O simulador tem a capacidade de realizar o reconhecimento de múltiplas entradas, conforme ilustrado na Figura 5. Para isso, deve-se seleccionar a opção de múltiplas entradas no menu Simulação->Múltiplas entradas. Na janela que se abrir, digitam-se as entradas desejadas na coluna Entrada e clica-se no botão “Simular”. Os resultados são exibidos na forma de realce colorido da entrada, da seguinte forma: verde (aceita), vermelho (rejeita) e amarelo (não foi possível simular (*loop infinito*)).

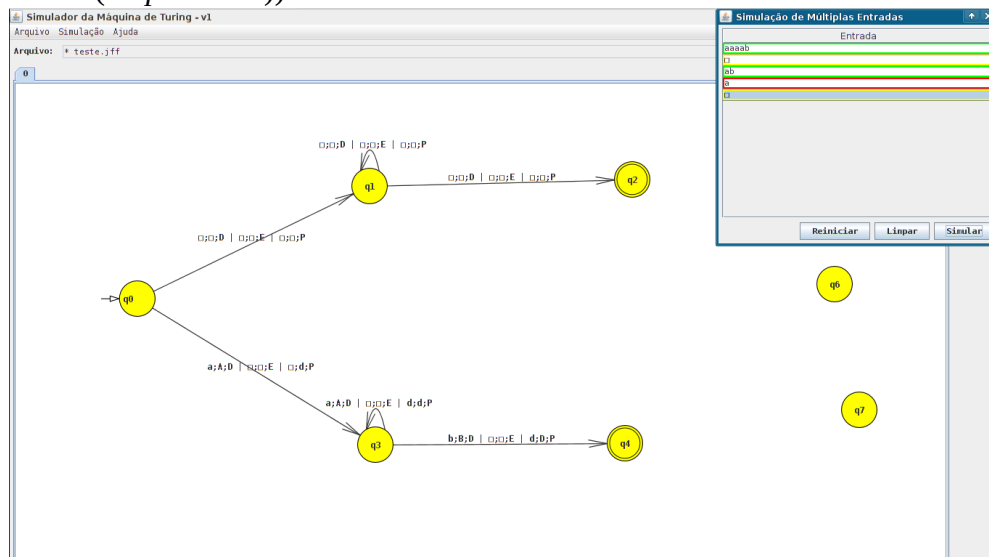


Figura 5: Múltiplas entradas.

3.4. Reconhecimento de uma entrada por meio de simulação passo a passo

O reconhecimento de entradas por meio de simulação passo a passo é possível através da opção do menu Simulação->Passo a passo. Na janela que se abrir, deve-se inserir uma entrada e, para realizar a simulação é necessário clicar no botão Play. Para pausar a simulação deve-se clicar no botão Pause, os demais botões servem para avançar ou retroceder o passo da simulação. A Figura 6 (em cima) ilustra esse processamento; quando um passo aceita ou rejeita a entrada, o *log* mostra o símbolo que foi aceito ou rejeitado na transição, que pode ser conferido na transição posicionando-se o cursor sobre. Já a Figura 6 (embaixo) ilustra o caso em que um *loop* infinito é “identificado” (não existe um algoritmo genérico que identifique um *loop* infinito em uma Máquina de Turing).

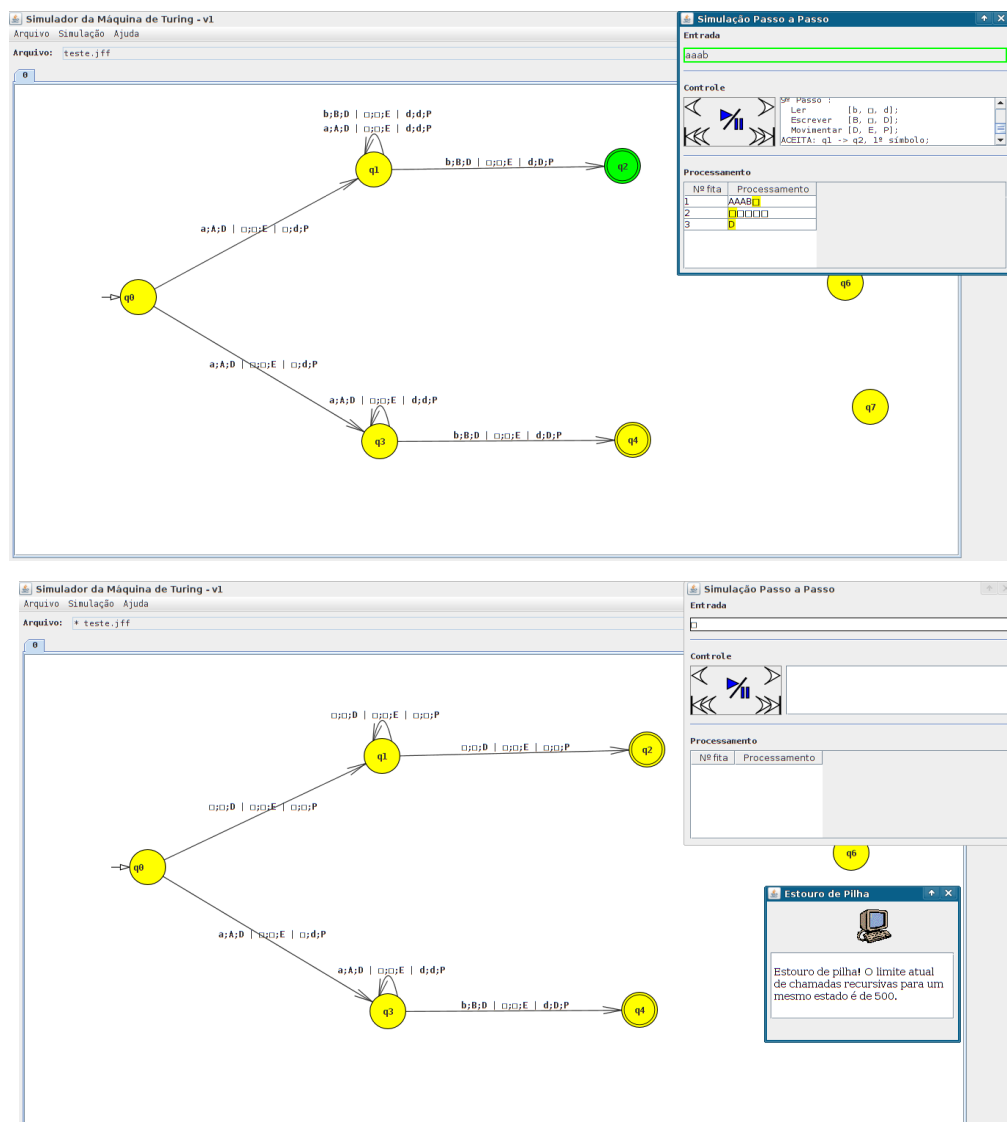


Figura 6: Simulação passo a passo: função computada aceita (em cima) e *loop* infinito (embaixo).

A condição de *loop* infinito ocorre em casos em que a fita atual de processamento é aceita infinitamente pra mesma transição ou conjunto de

transições. No caso da Figura 6 (embaixo) a entrada vazia entra em *loop* infinito no estado $q1$ pois, por mais que o estado $q2$ aceite a entrada, a transição de $q1$ para $q1$ vai “infinitamente” ler o símbolo de vazio e escrever o símbolo de vazio, já que as fitas de processamento são “infinitas” pra esquerda e pra direita. Essa identificação de *loop*, entretanto, pode ser errônea, visto que uma estrada pode ter caracteres que fazem com que a máquina percorra muitas vezes a mesma transição, e consigam terminar a simulação. Para resolver esse impasse, uma variável de limite de pilha foi criada, e pode ser configurada em Ajuda->Limite de pilha. Esse valor representa a quantidade máxima de vezes que um estado pode ser chamado recursivamente no mesmo *loop* de execução.

3.4. Abrir/Salvar uma MTM no mesmo formato que o JFlap (.jff)

Para permitir que as máquinas criadas sejam exportadas para serem abertas pelo JFlap ou por esse mesmo Simulador, foram desenvolvidas funcionalidades de abrir e salvar máquinas em formato JFlap Fyle (.jff), que podem ser acessadas através do menu Arquivo, como ilustra a Figura 7.

