Trabalho de Linguagens Formais, Autômatos e Computabilidade

Máquina de Turing

**Rafael Francisco Ferreira**

**Ciência da Computação – Universidade Estadual do Paraná (UNESPAR)**

**Apucarana – PR – Brasil**

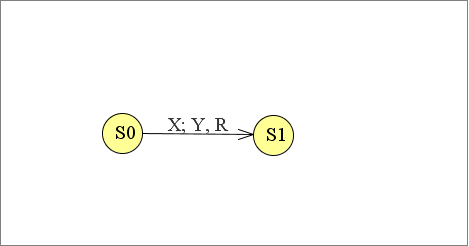
[rafaelfrancisco\_97@hotmail.com](mailto:rafaelfrancisco_97@hotmail.com)

**1. Introdução**

A Máquina de Turing é um dispositivo teórico conhecido como *máquina universal*, que foi concebido pelo matemático britânico Alan Turing (1912-1954), muitos anos antes de existirem os modernos computadores digitais. Num sentido preciso, é um modelo abstrato de um computador, que se restringe apenas aos aspectos lógicos do seu funcionamento (memória, estados e transições) e não à sua implementação física. Numa máquina de Turing pode-se modelar qualquer computador digital.

Uma máquina de Turing consiste em:

1. *Fita:* utilizada para leitura, rascunho e para escrita.
2. *Cabeçote:* mostra a posição atual da fita e se move para a direita e esquerda.
3. *Função de Transição:* função que movimenta a máquina a partir de um símbolo, gerando um novo símbolo, indo para um esta e movendo o cabeçote para a direita ou para a esquerda.
4. *Representação Gráfica:*



X: Símbolo lido da fita

Y: Símbolo escrito na fita

R: Sentido do movimento (Direita/Esquerda)

\*A fita é limitada à esquerda pelo marcador de início e infinita à direita, contendo o símbolo branco ao final da palavra.

Tipos de MT:

* Reconhecedora: responde se uma palavra pertence ou não à linguagem.
* Transdutora: é gerada uma palavra na própria fita que é a saída da MT.

Este relatório apresenta uma implementação de uma Máquina de Turing Reconhecedora e Transdutora.

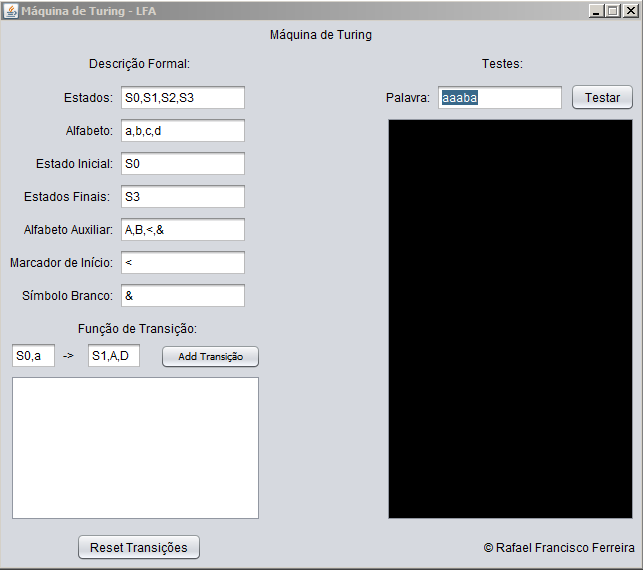
**2. Objetivo**

Esta implementação foi feita para um trabalho proposto na matéria de Linguagens Formais, Autômatos e Computabilidade.

A máquina tem a função de receber palavras e dizer se pertencem ou não à linguagem informada, ou mostrar as modificações feitas nas mesmas.

**3. A Máquina de Turing**

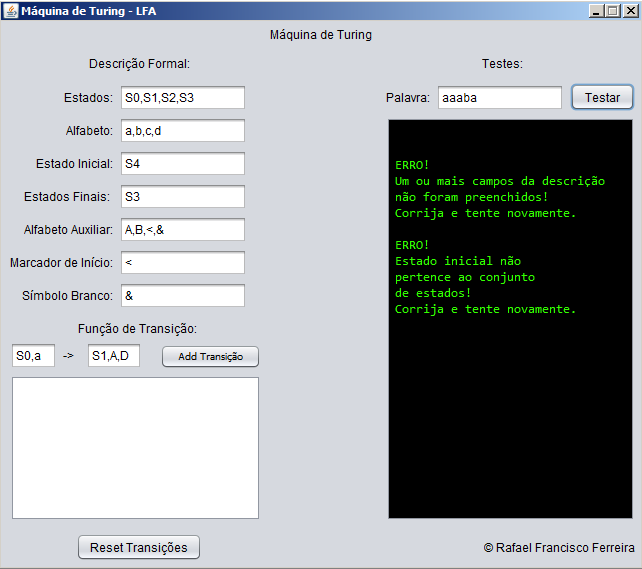
A máquina foi implementada de forma intuitiva, contendo campos para cada um dos itens da descrição formal.



Para utilizá-la, basta preencher os campos e adicionar a função de transição, feito isso, é só digitar uma palavra e clicar no botão “Testar”.

Caso haja um erro de digitação nos campos ou nas transições, um alerta poderá ser gerado informando o que o causou.

Se o usuário cometer um equívoco e adicionar uma transição com algum parâmetro errado, basta clicar em “Reset Transições”.



As mensagens de erro e alertas são mostrados na tela de testes sempre que o usuário tenta desempenhar uma ação e há algo de errado.

Se a máquina encontrar um erro durante o processamento da palavra, como um estado que não tem transição com a letra lida da fita, ela informará o erro, dizendo em qual estado parou e informar que a palavra não pertence à linguagem descrita.

Os campos são *case-sensitive,* ou seja, diferenciam letras maiúsculas de minúsculas. Deve-se estar atento ao adicionar alfabetos e nomear estados.

Para um melhor funcionamento da máquina, utilize sempre a vírgula para separar os nomes e símbolos dos alfabetos e conjuntos, sem espaços (!).

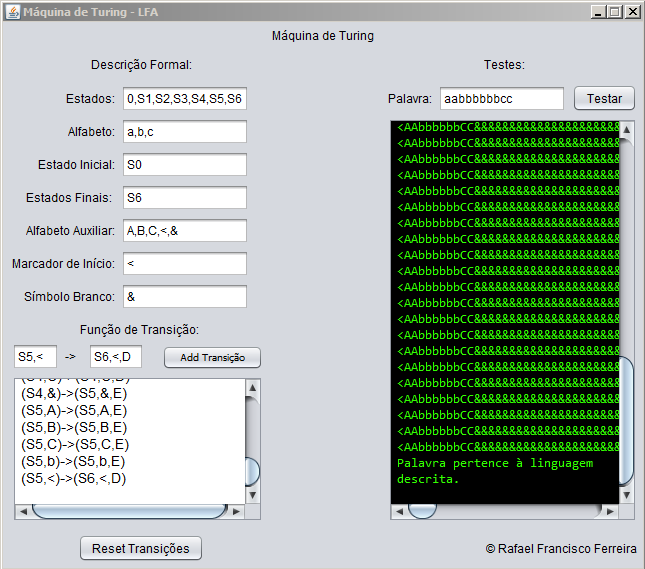
Ao abrir o programa, exemplos estarão pré-definidos nos campos de digitação. Caso queira mudar os símbolos, estes são escolha do usuário, mas lembre-se de manter a mesma formatação dos exemplos. Se algum símbolo digitado violar alguma regra ou condição, a máquina informará o erro.

**4. Exemplos**

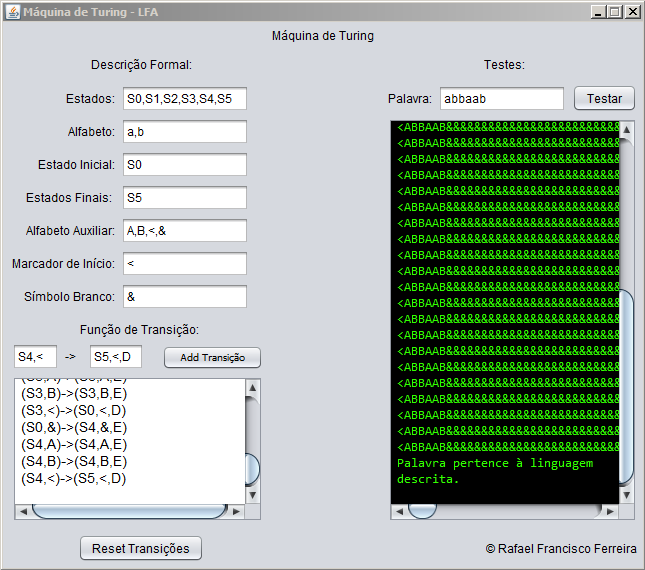
Abaixo estão alguns exemplos de utilização com linguagens aplicadas à máquina implementada.

**Reconhecedoras:**

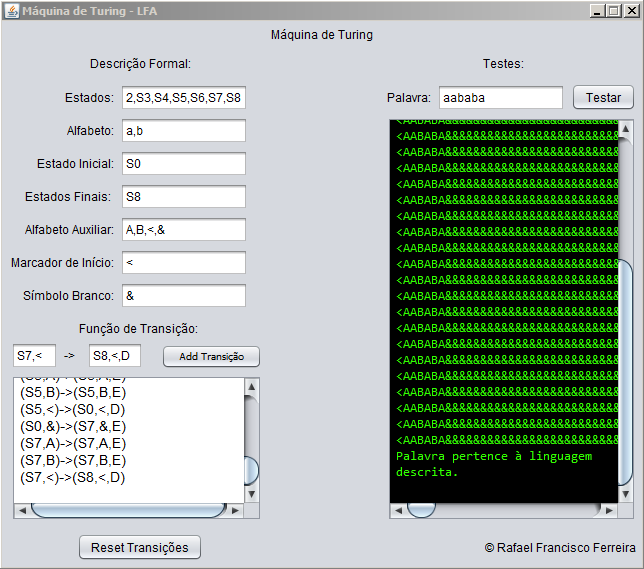
**L = {anbmcn / n>=0}**



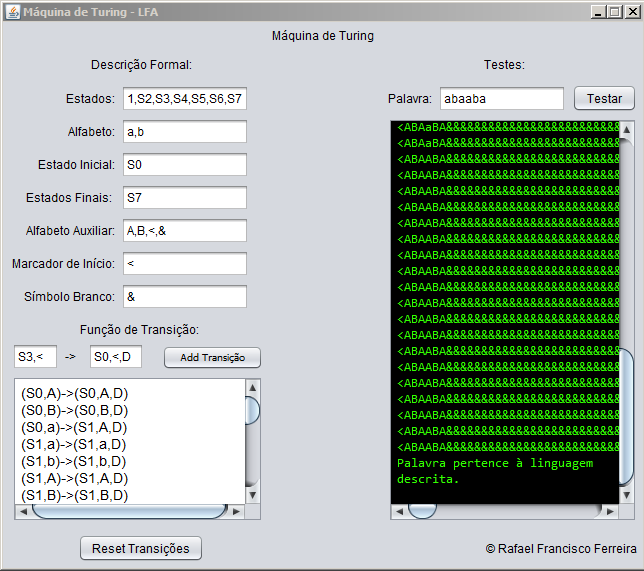
**L = {w ϵ {a,b}\*/ |w|a = |w|b}**



**L = {w ϵ {a,b}\*/ |w|a = 2\*|w|b}**

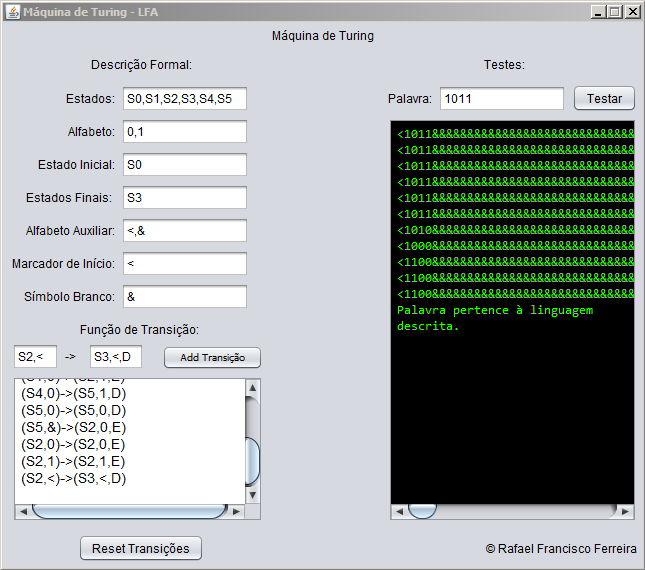


**L = {w ϵ {a,b}\*/ w é palíndromo}**

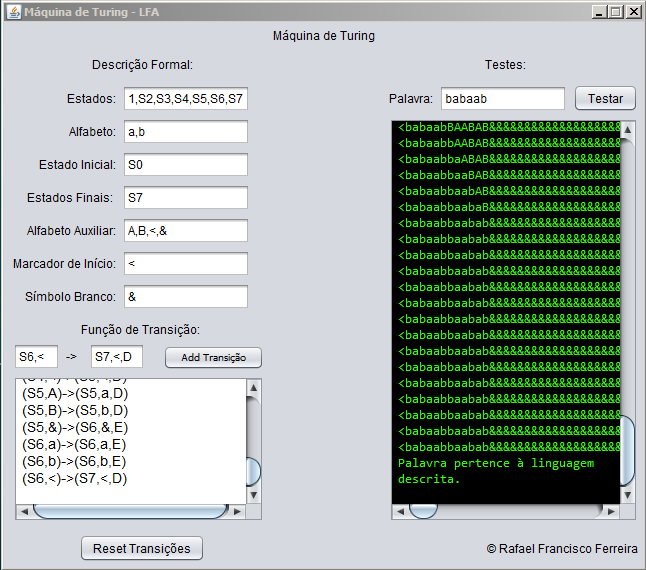


**Transdutoras**

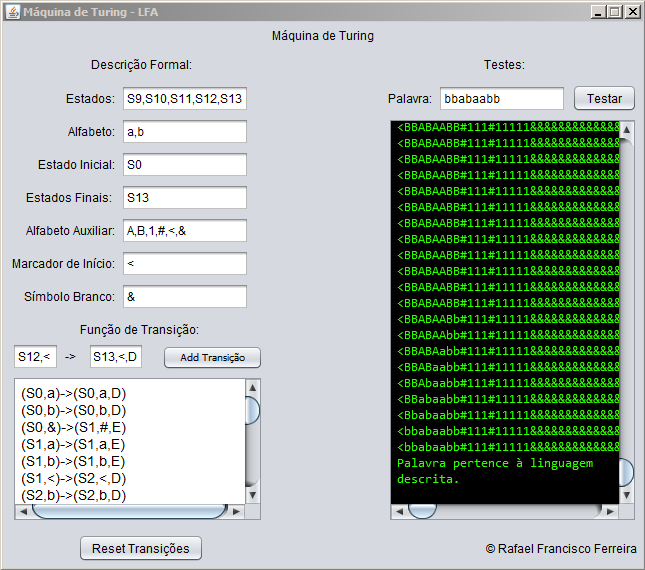
**L = Recebe um número binário e o incrementa em uma unidade:**

****

**L = Duplica qualquer palavra com a e b seguida dela mesma invertida (wwr)**

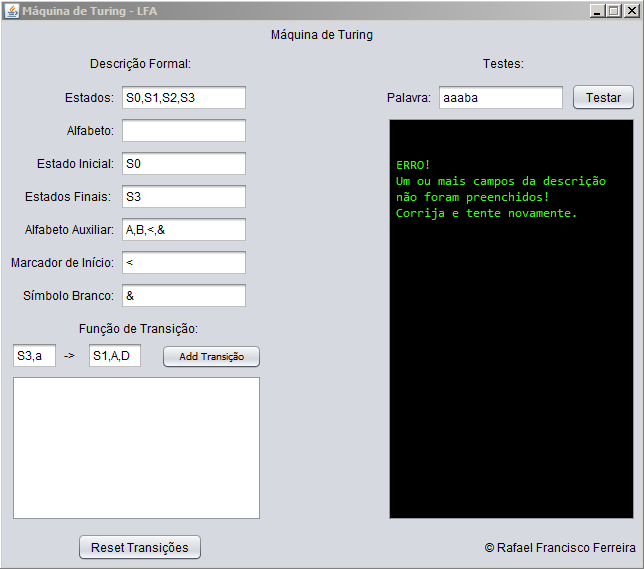
****

**L = Recebe uma palavra w com *a* e *b* e gera como saída w#qtdea#qtdeb**

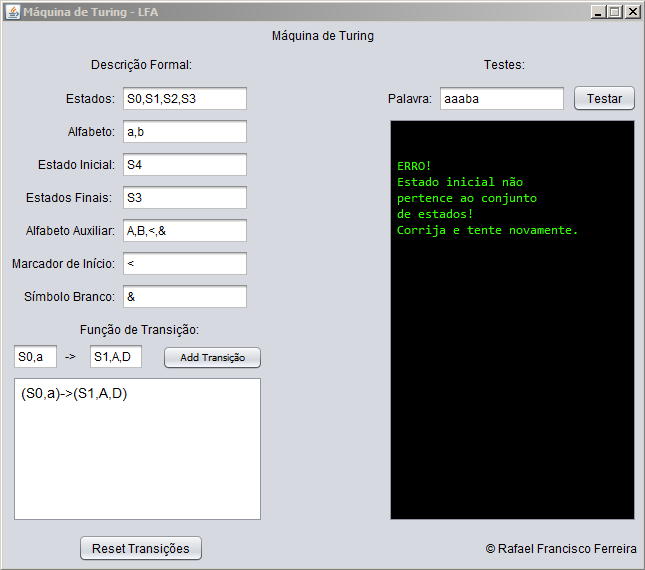
****

**Exemplos de Tratamento de Erro**

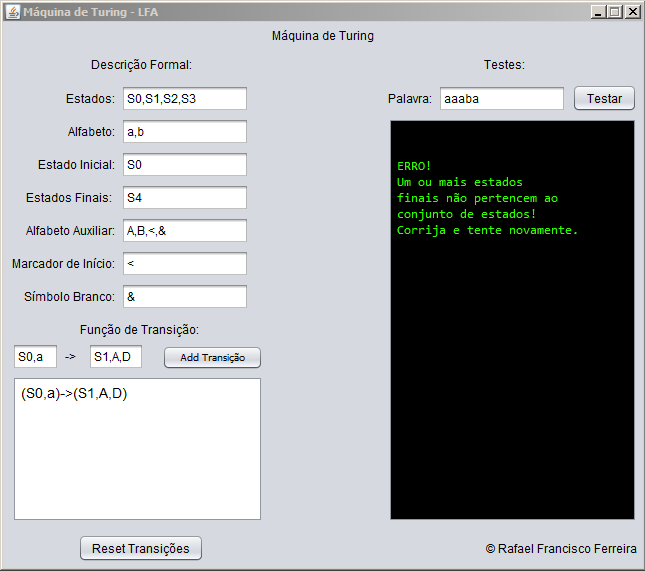
Caso o usuário esqueça um campo em branco:



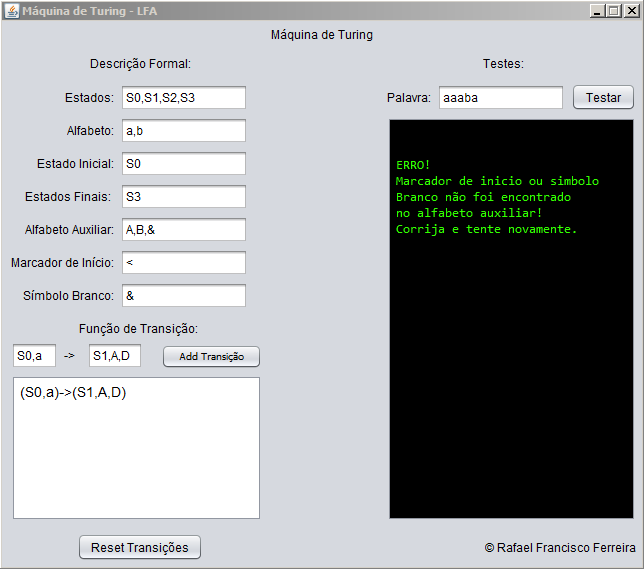
Caso o estado inicial não pertença ao conjunto de estados:



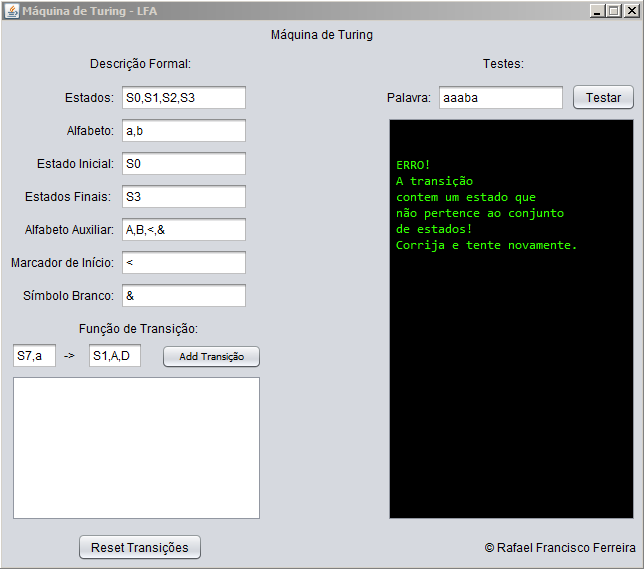
Caso um ou mais estados finais não pertençam ao conjunto de estados:



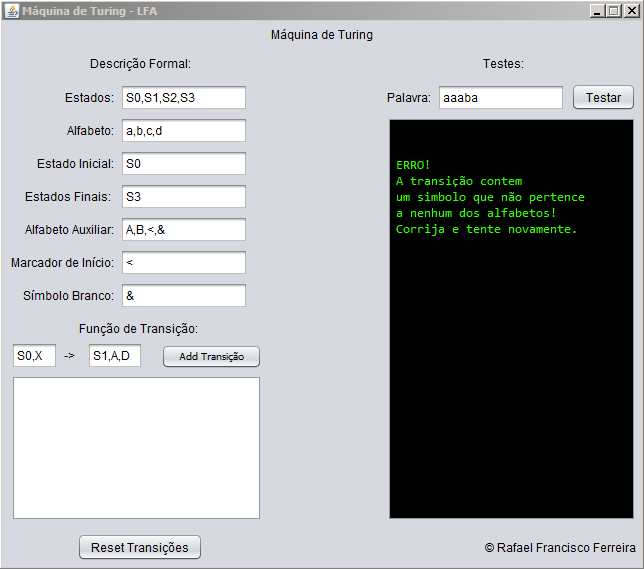
Caso o usuário esqueça de adicionar o marcador de inicio ou simbolo branco ao alfabeto auxiliar:



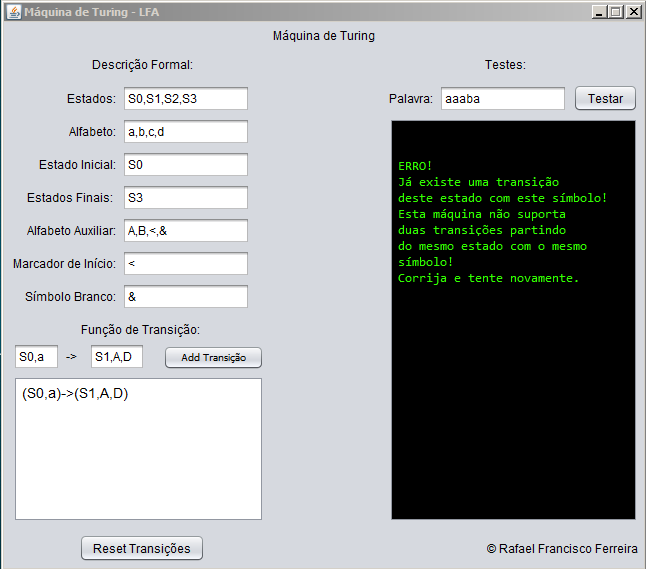
Caso a transição contenha um estado que não pertence ao conjunto de estados:



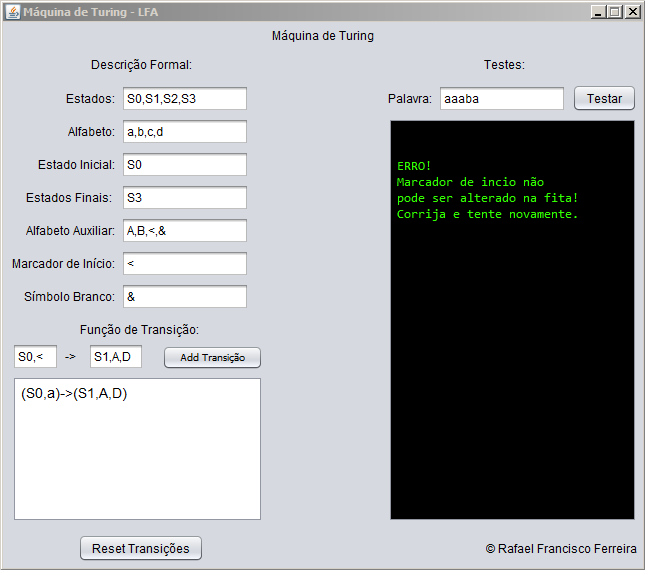
Caso a transição contenha um símbolo que não pertence aos alfabetos:



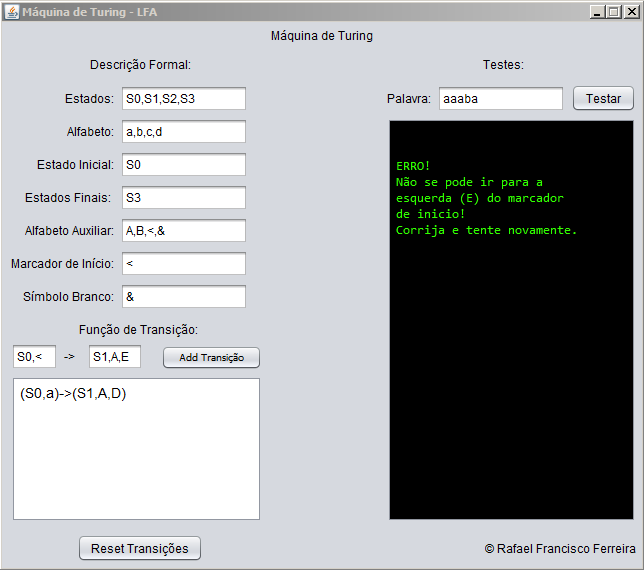
Caso o usuário tente criar duas transições partindo do mesmo estado com o mesmo símbolo:



Se o usuário tentar criar uma transição que alteraria o marcador de início:

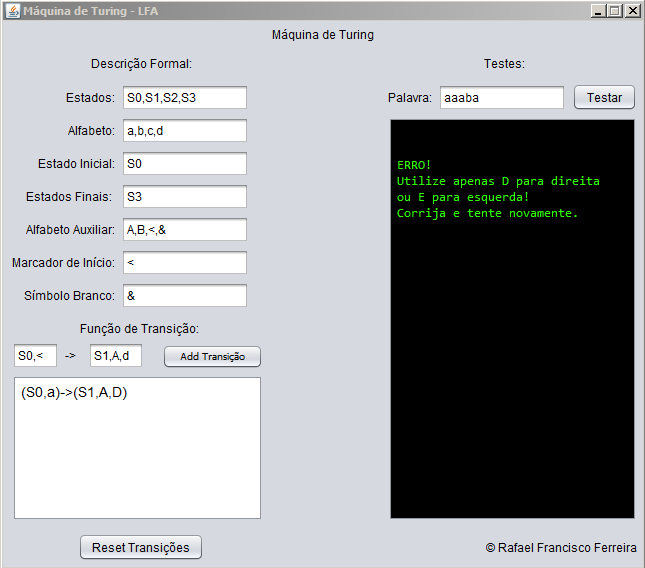


Se o usuário tentar criar uma transição que iria para a esquerda do marcador de início:

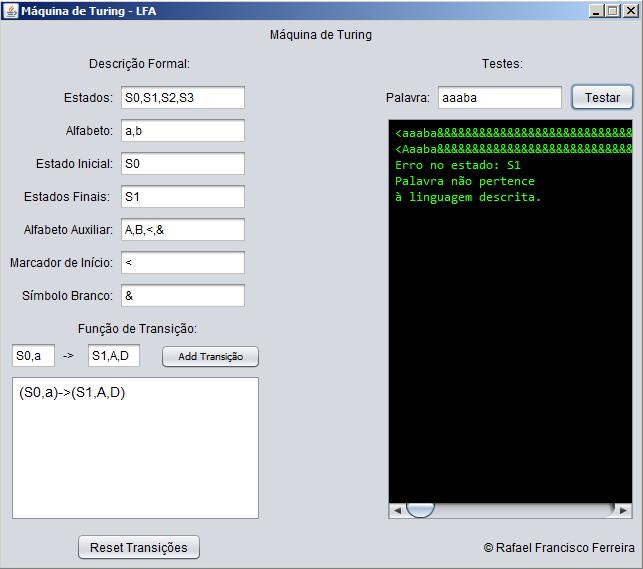


Caso o símbolo de direção esteja fora dos padrões:

(Estabelecidos como *D para direita e E para esquerda.*)



Caso a máquina encontre um estado de erro, ela informa em qual estado foi:



**5. Decisões de Projeto**

A primeira decisão importante foi a da linguagem de programação. A linguagem escolhida foi JAVA, levando em consideração, principalmente, que a mesma oferece uma grande variedade de opções para se trabalhar de forma prática com estruturas de dados, e que oferece a orientação a objetos.

As estruturas de dados utilizadas foram, em sua maioria, vetores, que armazenam informações como: alfabeto, alfabeto auxiliar, nomes dos estados, estados finais, palavra, partes das transições e etc. As outras estruturas utilizadas foram as Listas, que armazenam dados contidos em forma de objetos, tais como as Transições e os Estados, e também auxiliaram no armazenamento de Strings para tratamentos de erros e para consultas condicionais.

Os objetos foram criados para facilitar a implementação e organizar melhor o código . São objetos: MT, Fita, Transições, Estados. Cada objeto armazena suas respectivas informações privadas e públicas em estruturas de dados e variáveis locais. O objeto MT comanda todo o funcionamento da Máquina de Turing, sendo instanciado no momento que o usuário clica no botão “Testar”.

A interface com o usuário foi desenvolvida utilizando formulário java (JForm), visando facilitar o uso e também facilitar a captação dos dados.

Todos os erros foram tratados antes de se criar a máquina, exceto aqueles que a própria máquina deve informar, para evitar possíveis travamentos da máquina por erros de digitação ou campos mal preenchidos pelo usuário.

**6. Conclusão**

Este relatório apresentou uma implementação de uma Máquina de Turing Reconhecedora e Transdutora.

Como se nota nos exemplos, a máquina proposta foi testada com as mais diversas linguagens e retornou os resultados corretos em todos os testes feitos.

Além dos exemplos apresentados, duas listas de linguagens foram testadas mais de uma vez na máquina durante o desenvolvimento, e a mesma não falhou em nenhum dos testes.

Conclui-se, então, que a máquina implementada é eficaz no reconhecimento e transformação das palavras nela inseridas e testadas.

**7. Referências**

Vieira, Newton José.; (2006) “Introdução aos Fundamentos da Computação”,

Pioneira Thomson Learning.