UNIVERSIDADE ESTADUAL DO PARANÁ - *CAMPUS* APUCARANA

João Vitor de Souza Ribeiro

RELATÓRIO TÉCNICO - LFA

APUCARANA – PR
2023

João Vitor de Souza Ribeiro

RELATÓRIO TÉCNICO – LFA

Trabalho apresentado à disciplina de Linguagens Formais, Autômatos e Computabilidade do curso de Bacharelado em Ciência da Computação.

Professor: Guilherme Henrique de Souza Nakahata;

APUCARANA – PR
2023

SUMÁRIO

INTRODUÇÃO

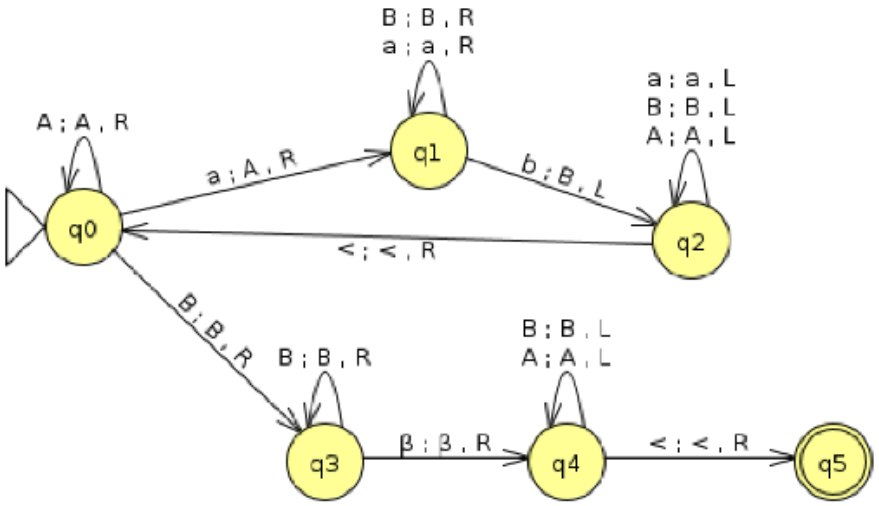
Diferente de outras disciplinas do curso de Ciência da Computação, Linguagens Formais e Autômatos possuem um grau de praticidade em suas atividades, de modo que os alunos possam realizar atividades “práticas” por meio da montagem de autômatos finitos, autômatos a pilha e máquinas de Turing, por exemplo; esta última sendo nicho principal de nossa análise e desenvolvimento nesta atividade. Através das aulas ministradas podemos entender que existem relações palpáveis entre uma máquina de Turing e a arquitetura e funcionamento dos computadores modernos, associando o processador ao cabeçote da fita, a fita à memória e os padrões de bits ao alfabeto da fita, por exemplo. De todo modo, se um problema não pode ser resolvido por uma MT, esse não poderá ser resolvido por qualquer sistema algoritmo.

Assim, cria-se um meio interessante para uma aplicação ainda mais prática de máquinas de Turing, uma vez que, possui relação com a criação algoritmica. A utilização de programação no trato de situações relacionadas à disciplina de LFA se mostra eficaz para associá-la a demais disciplinas e, também, para estruturar um conhecimento mais profundo e prático. Em exemplo, podemos citar uma atividade interdisciplinar registrada em uma IES Federal [2], onde, após o desenvolvimento de um programa que reproduzia o funcionamento da máquina de Turing, obteve um resultado positivo. Por se tratar de uma aplicação no primeiro semestre de uma formação acadêmica, cerca de 40% dos discentes relataram dificuldades para aprender uma linguagem e implementar o código nesta; no entanto, mesmo com as adversidades encontradas, apoximados 88% dos estudantes não caracterizaram o projeto como desistimulante.

Em geral, há questões pertinentes que façam uma aplicação prática da disciplina, especialmente de Máquinas de Turing ser altamente proveitosa e vantajosa para aprendizados interdisciplinares, além de uma possível maior absorção do cointeúdo ministrado em sala, por conta do grau maior de dedicação para a criação e funcionamento de um código.

CAPÍTULO 1
OBJETIVOS

O objetivo principal do código criado pode ser interpretado como estabelecer o funcionamento prático de uma máquina de Turing, mas, em um programa. De modo geral, o código fonte tem como objetivo receber uma descrição formal, ou seja, as características de um conjunto de estados que represente uma máquina de Turing, com isso, poderá avaliar a aprovação ou não de palavras mediante a linguagem informada por meio das transições. A MT também deve ser capaz de modificar a fita, a fim de que as transdutoras sejam ainda passíveis de verificação. Um exemplo de máquina de Turing do qual o algoritmo deve ser capaz de reconhecer a linguagem e verificar o pertencimento de palavras está representado na Figura 1.



	a	b	A	B	<	β
q0	q1,A,R	X	q0,A,R	q3,B,R	X	X
q1	q1,a,R	q2,B,L	X	q1,B,R	X	X
q2	q2,a,L	X	q2,A,L	q2,B,L	q0,<,R	X
q3	X	X	X	q3,B,R	X	q4,β,L
q4	X	X	q4,A,L	q4,B,L	q5,<,R	X
q5	X	X	X	X	X	X

Figura 1

CAPÍTULO 2
MOTIVAÇÃO E RECURSOS UTILIZADOS

Baseando-se no exposto anteriormente, devemos explicitar os motivos para a realização do trabalho, ou seja, o objetivo final e os recursos utilizados para que isso seja cumprido.

2.1 Motivação

Como citado no capítulo que trata acerca dos objetivos do projeto em questão, a motivação também seria a realização de um código fonte funcional em que possamos demonstrar o pleno funcionamento de uma máquina de Turing, ao passo que recebemos as informações necessárias e poderemos determinar se as palavras requisitadas fazem ou não parte daquela linguagem, além de modificar a fita nos casos em que a Máquina de Turing for do tipo transdutora. Para tal, precisamos receber os dados, tais como a descrição formal e a tabela de transições, após, é necessário que se crie uma lógica de programação para atingir-se o esperado, de maneira eficaz. Na figura 2, por exemplo, podemos visualizar como a descrição formal deverá ser recebida, de maneira a imprimir o seu conteúdo formatado no modelo contido nesta.

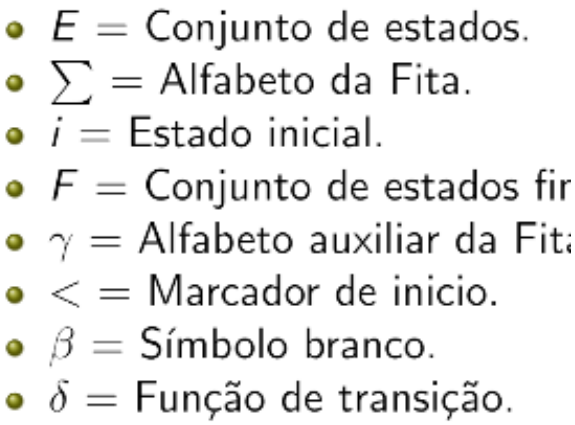


Figura 2

Assim, faz-se necessário, com as informações pertinentes acerca dos objetivos e motivações, esmiuçar-se àqueles dados relevantes à Estrutura de Dados, Linguagem de Programação e demais questões acerca da implementação do código em questão.

2.2 Estrutura de Dados

Sob uma perspectiva geral, o código fonte é trabalhado com uma estrutura composta pela criação de uma matriz que represente a conexão de um estado e cada uma das letras ou símbolos do alfabeto geral da MT. Essa matriz deveria conter três dados pertinentes para o funcionamento da máquina, sendo eles: estado futuro, alfabeto futuro e direção. Em casos que o campo for anulado os três passariam a ser vistos como x, para uma melhor visualização.

Visando um melhor aproveitamento e funcionamento, uma classe *Transições* foi criada para implementar um vetor de 3 posições, responsável por alocar os três dados mencionados, ambos no formato String. A partir dessa classe, podemos criar a matriz citada, intitulada *mat [ ] [ ],* do tipo Objeto de *Transições*, ou seja, em cada posição da matriz teremos um vetor de três dados.

Para o recebimento dos dados iniciais, tais como a descrição formal, alguns vetores foram criados, como *alfabeto [ ],* que armazena as letras do alfabeto principal, *estados [ ],* que armazena os estados baseado no estado inicial até a quantidade total de estados e *finais [ ],* que armazena o/s estado/s final/is do exemplo. Ainda no recebimento dos dados a matriz que citamos acima é criada, recebendo as transições necessárias para estabelecer a linguagem da máquina de Turing; para cada uma das relações entre um estado e letra do alfabeto é requisitado que o usuário digite um estado futuro no formato completo – por exemplo q0 –, o alfabeto futuro e a direção – D ou E.

Olhando para a parte da execução, o código conta com a criação de um vetor *alfabetogeral [ ],* que une em um só vetor as letras presentes no alfabeto principal, o marcador de início e o símbolo de branco, assim facilitando a conferência baseado na fita. Para que possamos executar de fato nosso programa, necessitamos de ao menos uma palavra para teste, esta, quando recebida, é inserida em um vetor *vet [ ]*, antecedida de um marcador de início e precedida de brancos até que o vetor esteja completo. Essa palavra será utilizada para uma conferência no vetor *alfabetogeral [ ],* a partir do momento em que esta letra for encontrada no vetor, o índice dele passa a ser um “guia” para verificar, baseado no estado atual, quais as transições futuras para aquela situação.

A execução é mantida até que todas as transições tenham sido avaliadas, de modo com que o while (repetição) seja quebrada, assim verificando se a palavra foi ou não aceita. Para isso, observamos se o último estado acessado era um estado final, se for o caso, a palavra é aceita, do contrário, a palavra é rejeitada. Outras estruturas de repetição e “organização” foram utilizadas no programa, mas somente de modo a facilitar e possibilitar o funcionamento geral.

2.3 Linguagem de Programação e demais informações

A linguagem escolhida para a implementação do código foi Java, de maneira que se utiliza-se a linguagem principalmente abordada no segundo ano de formação para implementar o ciclo de instruções. Java é uma linguagem de alto nível desenvolvida pela Sun Microsystems, orientada a objetos e amplamente utilizada em vários ramos da programação. Inclusive, no código é utilizado esta orientação a objetos; a única biblioteca da linguagem utilizada para o funcionamento do código foi o Scanner, para recebimento dos dados pelo usuário, via console.

CAPÍTULO 3
RESULTADOS

Mediante os objetivos apresentados, o resultado esperado seria o pleno funcionamento de um código que exemplifique como uma máquina de Turing reconhece ou não palavras de uma determinada linguagem dado a sua descrição formal e tabela de transições. Dessarte, com a implementação completa e revisada, os resultados foram atingidos, resultando em uma aplicação interativa e funcional, recebendo a quantidade de letras no alfabeto, as letras, uma letra para representar os estados, a quantidade de estados, a quantidade de estados finais, o estado inicial, os estados finais, marcador de início, símbolo de branco e as transições contendo estado futuro, alfabeto futuro e direção (quando não anulado). Abaixo, nas Figuras 3,4,5,6,7 e 8, podemos ver o funcionamento do exemplo de máquina de Turing representado na Figura 1.

```
============================== MÁQUINA DE TURING ==============================
Digite a quantidade de letras do alfabeto: 4
Informe a letra 1 do alfabeto: a
Informe a letra 2 do alfabeto: b
Informe a letra 3 do alfabeto: A
Informe a letra 4 do alfabeto: B
Informe uma letra para representar os estados: q
Digite a quantidade total de estados: 6
Digite o estado inicial - somente o nº -: 0
Digite a quantidade de  estados finais: 1
Digite o 1º estado final - somente o nº -:  5
Digite o marcador de início: <
Digite o marcador de branco: *
****************************** Descrição Formal ******************************
Σ = {a,b,A,B}
E = {q0,q1,q2,q3,q4,q5}
i =  q0
F = {q5}
γ =  {<,*}
< =  <
β =  *
**************************** Tabela de Transição ****************************
δ     a     b     A     B     <     *
q0   1,1   1,2   1,3   1,4   1,5   1,6
q1   2,1   2,2   2,3   2,4   2,5   2,6
q2   3,1   3,2   3,3   3,4   3,5   3,6
q3   4,1   4,2   4,3   4,4   4,5   4,6
q4   5,1   5,2   5,3   5,4   5,5   5,6
q5   6,1   6,2   6,3   6,4   6,5   6,6
********************************* Transições *********************************
Obs: Se não existir uma transição, insira x
Obs: Uma transição inválida fará com que o campo seja anulado
----------------------------- Digite as transições -----------------------------
--------------------------------------------------------------------------------
Digite o estado futuro da transição {1,1}: q1
Digite o alfabeto futuro da transição {1,1}: A
Digite a direção da transição {1,1}: R
A direção informada não é valida, digite D ou E: D
--------------------------------------------------------------------------------
```

Figura 3

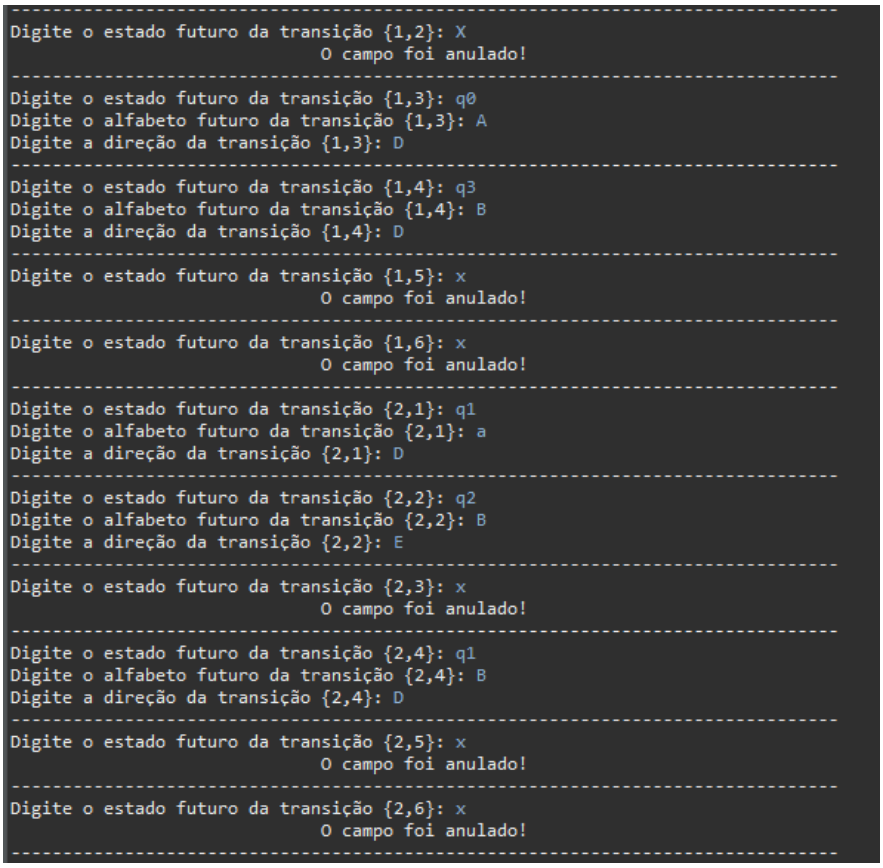


Figura 4

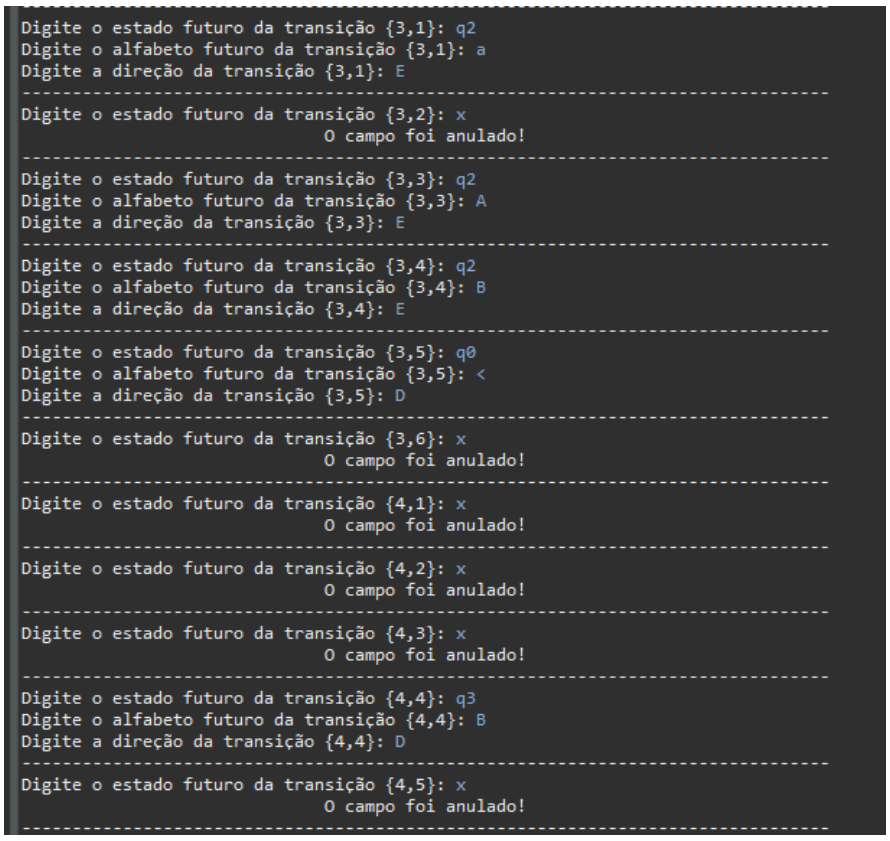


Figura 5

```
Digite o estado futuro da transição {4,6}: q4
Digite o alfabeto futuro da transição {4,6}: *
Digite a direção da transição {4,6}: E
--------------------------------------------------------------------------------
Digite o estado futuro da transição {5,1}: x
                            O campo foi anulado!
--------------------------------------------------------------------------------
Digite o estado futuro da transição {5,2}: x
                            O campo foi anulado!
--------------------------------------------------------------------------------
Digite o estado futuro da transição {5,3}: q4
Digite o alfabeto futuro da transição {5,3}: A
Digite a direção da transição {5,3}: E
--------------------------------------------------------------------------------
Digite o estado futuro da transição {5,4}: q4
Digite o alfabeto futuro da transição {5,4}: B
Digite a direção da transição {5,4}: E
--------------------------------------------------------------------------------
Digite o estado futuro da transição {5,5}: q5
Digite o alfabeto futuro da transição {5,5}: <
Digite a direção da transição {5,5}: D
--------------------------------------------------------------------------------
Digite o estado futuro da transição {5,6}: x
                            O campo foi anulado!
--------------------------------------------------------------------------------
Digite o estado futuro da transição {6,1}: x
                            O campo foi anulado!
--------------------------------------------------------------------------------
Digite o estado futuro da transição {6,2}: x
                            O campo foi anulado!
--------------------------------------------------------------------------------
Digite o estado futuro da transição {6,3}: x
                            O campo foi anulado!
--------------------------------------------------------------------------------
Digite o estado futuro da transição {6,4}: x
                            O campo foi anulado!
--------------------------------------------------------------------------------
Digite o estado futuro da transição {6,5}: x
                            O campo foi anulado!
```

Figura 6

```
--------------------------------------------------------------------------------
Digite o estado futuro da transição {6,6}: x
                            O campo foi anulado!
**************************** Tabela de Transição ****************************
δ     a        b        A        B        <        *
q0   q1,A,D  Sx,x,x  q0,A,D  q3,B,D  Sx,x,x  Sx,x,x
q1   q1,a,D  q2,B,E  Sx,x,x  q1,B,D  Sx,x,x  Sx,x,x
q2   q2,a,E  Sx,x,x  q2,A,E  q2,B,E  q0,<,D  Sx,x,x
q3   Sx,x,x  Sx,x,x  Sx,x,x  q3,B,D  Sx,x,x  q4,*,E
q4   Sx,x,x  Sx,x,x  q4,A,E  q4,B,E  q5,<,D  Sx,x,x
q5   Sx,x,x  Sx,x,x  Sx,x,x  Sx,x,x  Sx,x,x  Sx,x,x
********************************************************************************
```

Figura 7

```
********************************************************************************
Digite uma palavra para ser testada: aaabb
********************************* FITA INICIAL *********************************
< a a a b b * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
********************************************************************************
********************************* FITA FINAL ***********************************
< A A A B B * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
********************************************************************************
A palavra não foi aceita!
Deseja testar outra palavra? 1 - SIM | 2 - NÃO
Opção Escolhida: 1
Digite uma palavra para ser testada: aabb
********************************* FITA INICIAL*********************************
< a a b b * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
********************************************************************************
********************************* FITA FINAL ***********************************
< A A B B * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
********************************************************************************
A palavra foi aceita!
```

Figura 8

CONCLUSÃO

Com base no exposto anteriormente, podemos evidenciar que uma aplicação prática, mesmo que para uma disciplina mais prática que outras, pode ser vantajoso e apropriado para a obtenção de conhecimentos externos a disciplina e, também, para uma maior fixação do conteúdo visto em sala. A atividade interdisciplinar citada anteriormente [2], se faz uma prova que disciplinas, por mais distantes que sejam, superficialmente observadas, podem desempenhar relações importantes, como é o exemplo de LFA e POO, no caso desta implementação em específico, pelo uso de Objetos em Java.

De um modo geral, métodos diferenciados de ensino, por mais “simples” que sejam, podem resultar em influências palpáveis na conexão entre estudantes e o conhecimento. Mazur (2015) [1], estabeleceu que estes métodos distintos, conhecidos como metodologias ativas, diminuem a abstração e aproxima estudantes de uma resolução mais íntima dos problemas, uma vez que procuram por si mesmos um meio de resolução, ao passo que podem se comunicar e auxiliar seus colegas, transmitindo conhecimentos de formas pouco possíveis entre discentes e docentes. Em suma, o código apresentado desempenhou o papel esperado, tanto em funcionamento, quanto em objeto de aprendizado ativamente diferenciado.

REFERÊNCIAS

[1] MAZUR, Eric. **Peer Instruction -A Revolução da Aprendizagem Ativa**.Editora Penso. Ano 2015.

[2] SANTANA, J. S. Ícaro; NASCIMENTO, M. S. Flávia; BEZERRA, M. S. Romildo; BRITO. Rodrigo; OLIVEIRA, Bruno. **Utilizando a Máquina de Turing para o Desenvolvimento de um Projeto Interdisciplinar**. (2015) Disponível em:<https://www.researchgate.net/publication/332423755_Utilizando_a_Maquina_de_Turing_para_o_Desenvolvimento_de_um_Projeto_Interdisciplinar>.