UNIVERSIDADE ESTADUAL DO PARANÁ – UNESPAR
CIÊNCIA DA COMPUTAÇÃO – 2º ANO

Guilherme Alves da Silva

TRABALHO DE LINGUAGENS FORMAIS, AUTÔMATOS E COMPUTABILIDADE
Máquina de Turing

APUCARANA – PR
2017

Introdução

Na matemática, a computabilidade é algo frequente e natural pois as aplicações para fórmulas podem ser das mais variadas e em várias circunstâncias, significando que cálculos precisam manter uma baixa margem de erro. Engenheiros são os profissionais encarregados de desenvolver e manter os sistemas e mecanismos que geralmente funcionam respeitando as fórmulas devenvolvidas por matemáticos ou outros cientistas. A necessidade de uma ferramenta para auxiliar nos cálculos apareceu muitos anos atrás, como por exemplo na Mesopotâmia foi desenvolvido o ábaco, uma ferramenta computacional que provou sua utilidade.

Na atualidade a ferramenta computacional mais utilizada é o computador, uma máquina digital programável capaz de processar automaticamente dados ou informação, respeitando algoritmos implantados por meio de programas usando linguagens de programação, que também respeitam a arquitetura de cada computador.

Computadores digitais certamente são encontrados com facilidade atualmente, mas os computadores humanos foram cruciais e muito utilizados na Segunda Guerra Mundial. Um computador humano basicamente significa uma pessoa que computa, realiza cálculos matemáticos, algo que foi necessário antes que computadores eletrônicos estivessem comercialmente disponíveis.

Com a vasta disponibilidade de computadores digitais e aplicações para o mesmo, um campo específico para este estudo surgiu naturalmente, chamado de ciência da computação.

Justificativa

Teoria dos autômatos é o estudo de máquinas abstratas, objetos matemáticos, dentro da ciência da computação. Considerando os problemas que podem ser resolvidos usando esses objetos, uma relação com o campo da teoria das linguagens formais se desenvolveu.

O campo das linguagens formais é dedicado ao estudo que possibilita, usando modelos matemáticos, o reconhecimento e a especificação de linguagens, falando amplamente de suas estruturas, propriedades, classificações, características e inter-relacionamentos.

Tratando aspectos teóricos da ciência da computação, como por exemplo complexidade computacional, computabilidade e decidibilidade, a teoria das linguagens formais também ajuda na modelagem de sistemas, processamento de linguagens e reconhecimento de padrões.

Considerando uma linguagem formal, neste campo é possível representá-la de maneira precisa e finita com uma sustentação matemática, que pode ser feita através de reconhecedores e geradores. Um dispositivo reconhecedor serve para verificar se uma palavra pertence ou não à uma determinada linguagem, como por exemplo fazendo uso dos autômatos, temos os autômatos com pilha, e a máquina de Turing que é uma ferramenta útil para testar e desenvolver algoritmos, também sendo usada no desenvolvimento de compiladores.

Máquina de Turing

Desenvolvida pelo matemático britânico Alan Turing em 1936, anos antes de existir qualquer computador digital, a máquina de Turing é um dispositivo teórico e abstrato que representa um computador, considerando somente alguns aspectos teóricos de seu funcionamento como por exemplo uma memória, estados e transições. A máquina também é conhecida como computador ou máquina universal.

Essa máquina abstrata, que é um modelo de computação matemática com grande poder computacional, manipula símbolos em uma fita de acordo com uma tabela de regras. Seguindo a lógica de um algoritmo na construção da tabela, a máquina consegue simular o funcionamento de tal algoritmo.

Funcionando em uma fita teoricamente infinita de memória, que está dividida em células, a máquina posiciona sua cabeça, que se move, em uma determinada célula para ler o símbolo que está armazenado em tal célula. Então a máquina consulta sua tabela de funções de transição, e considerando o símbolo que está armazenado na célula, juntamente com o estado em que a máquina se encontra em tal momento, realiza uma ação de apagar o símbolo da fita, escrever na fita o símbolo que está armazenado na tabela, mudar para outro estado e mover a cabeça na fita para a esquerda ou direita. Porém, existem alguns tipos diferentes de máquinas de Turing, portanto essas ações não se aplicam para todos os tipos existentes. Algumas variações das máquinas de Turing envolvem ações como poder deixar a cabeça imóvel na fita, usar múltiplas fitas, fitas ilimitadas para a esquerda, e outras variações.

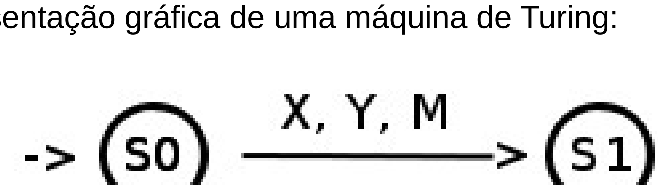
A máquina de Turing reconhecedora é capaz de determinar se uma palavra pertence ou não à uma linguagem, tendo como saída a resposta sim ou não. Porém a máquina de Turing transdutora gera uma palavra na própria fita, que é a saída desse tipo de máquina.

Considerando uma máquina de Turing padrão, ela basicamente é constituída por uma fita que é usada para leitura, rascunho, escrita e como memória, um cabeçote que mostra a posição atual e que se movimenta na fita, um registrador de estados que armazena o estado em que a máquina está em um determinado momento na sua execução, e uma tabela de função de transição que armazena todas as instruções sobre o que fazer na fita.

O cabeçote na fita começa na posição ao lado direito do marcador de início da fita, ao terminar o processamento da palavra ele pode parar na mesma posição, mas não é obrigatório.

Para reconhecer ou traduzir uma palavra, a máquina deve processá-la e parar em algum estado final.

Representação gráfica de uma máquina de Turing:



X = Símbolo lido na fita;

Y = Símbolo escrito na fita;

M = Sentido do movimento do cabeçote na fita (esquerda ou direita);

S0, **S1** = Estados;

A **seta** entre os círculos representa uma transição entre os estados S0 e S1;

A pequena seta chegando no estado S0 indica um estado inicial.

Um estado final é representado por um círculo duplo em um determinado estado.

<	a	a	a	b	b	b	β	β	β

^

A fita é limitada à esquerda pelo marcador de início de fita “<”, e ilimitada à direita com “β” sendo usado para representar partes da fita em que a palavra já terminou. A palavra presente na fita acima é “aaabbb”.

A descrição formal de uma máquina de Turing é composta de:

MT = { **Σ**, **E**, **i**, **F**, **Γ**, **<**, **β**, **δ** };

Σ = Alfabeto finito da fita (de entrada);

E = Conjunto finito de estados;

i = Estado inicial ($i \in E$);

F = Conjunto de estados finais ($F \subset E$);

Γ = Alfabeto auxiliar da fita;

< = Marcador de início da fita ($< \in \Gamma$);

β = Símbolo que representa espaços em branco na fita ($\beta \in \Gamma$);

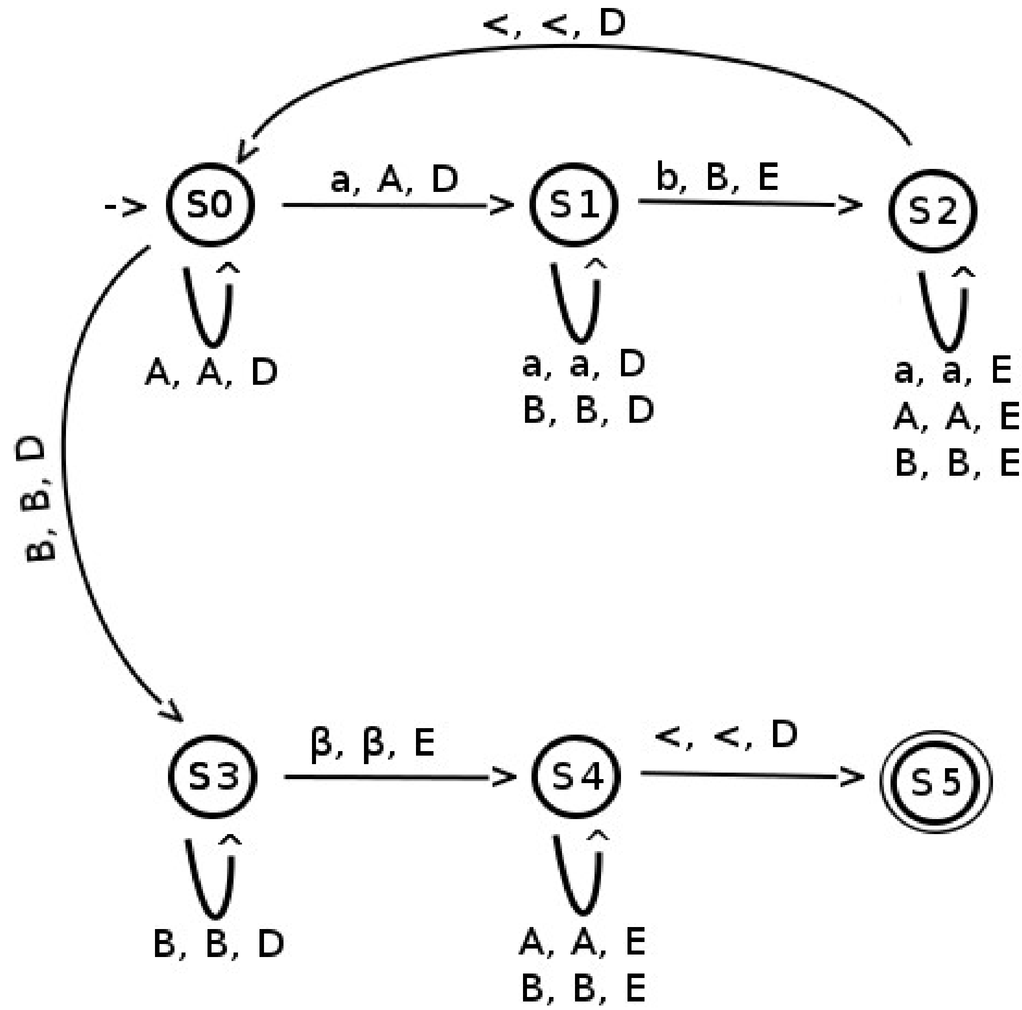
δ = Função de transição.

$$\mathbf{E \times (\Sigma \cup \Gamma) \rightarrow E \times (\Sigma \cup \Gamma) \times (E , D)}$$

Exemplos

Máquina de Turing Reconhecedora

L = { a^n b^n / n > 0 }



Σ = { a, b };

E = { S_0 , S_1 , S_2 , S_3 , S_4 , S_5 };

i = S_0 ;

F = { S_5 };

Γ = { <, β, A, B };

< = < ;

β = β ;

δ =

	Σ		Γ			
	a	**b**	**<**	**β**	**A**	**B**
S_0	S_1 , A, D	X	X	X	S_0 , A, D	S_3 , B, D
S_1	S_1 , a, D	S_2 , B, E	X	X	X	S_1 , B, D
S_2	S_2 , a, E	X	S_0 , <, D	X	S_2 , A, E	S_2 , B, E
S_3	X	X	X	S_4 , β, E	X	S_3 , B, D
S_4	X	X	S_5 , <, D	X	S_4 , A, E	S_4 , B, E
S_5	X	X	X	X	X	X

Testando a palavra “aaabb”:

```
        |    a    |    b    |    <    |    β    |    A    |    B    |
   -----------------------------------------------------------------
    000 | 001,A,D |    X    |    X    |    X    | 000,A,D | 003,B,D |
   -----------------------------------------------------------------
 >  001 | 001,a,D | 002,B,E |    X    |    X    |    X    | 001,B,D |
   -----------------------------------------------------------------
    002 | 002,a,E |    X    | 000,<,D |    X    | 002,A,E | 002,B,E |
   -----------------------------------------------------------------
    003 |    X    |    X    |    X    | 004,β,E |    X    | 003,B,D |
   -----------------------------------------------------------------
    004 |    X    |    X    | 005,<,D |    X    | 004,A,E | 004,B,E |
   -----------------------------------------------------------------
    005 |    X    |    X    |    X    |    X    |    X    |    X    |
   -----------------------------------------------------------------


< A A A B B β β
            ^
A palavra nao foi aceita.
```

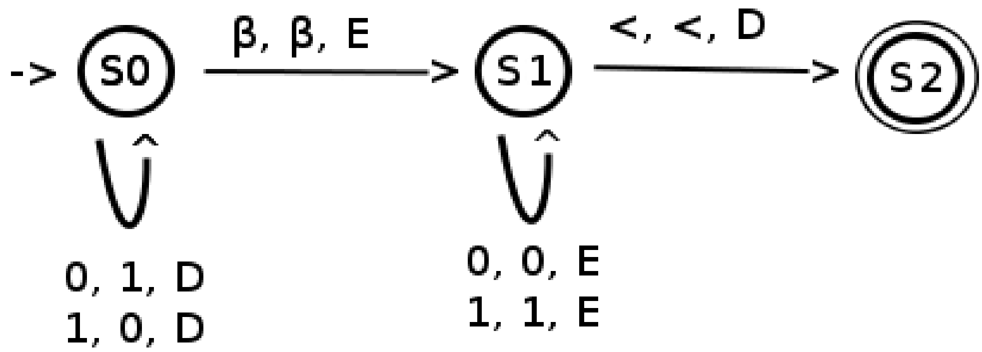
Testando a palavra “aaabbb”:

	a	b	<	β	A	B
000	001,A,D	X	X	X	000,A,D	003,B,D
001	001,a,D	002,B,E	X	X	X	001,B,D
002	002,a,E	X	000,<,D	X	002,A,E	002,B,E
003	X	X	X	004,β,E	X	003,B,D
004	X	X	005,<,D	X	004,A,E	004,B,E
> 005	X	X	X	X	X	X

```
< A A A B B B β β
  ^
A palavra foi aceita.
```

Máquina de Turing Transdutora

L = Tem como entrada um número binário e gera como saída o complemento do número binário.



Σ = { 0, 1 };

E = { S_0 , S_1 , S_2 };

i = S_0 ;

F = { S_2 };

Γ = { <, β };

< = < ;

β = β ;

δ =

	Σ		**Γ**	
	0	**1**	**<**	**β**
S_0	S_0 , 1, D	S_0 , 0, D	X	S_1 , β , E
S_1	S_1 , 0, E	S_1 , 1, E	S_2 , <, D	X
S_2	X	X	X	X

Tendo como entrada “0100101”, o resultado seria:

```
     |    0    |    1    |    <    |    β    |
  ---------------------------------------------
  000 | 000,1,D | 000,0,D |    X    | 001,β,E |
  ---------------------------------------------
  001 | 001,0,E | 001,1,E | 002,<,D |    X    |
  ---------------------------------------------
> 002 |    X    |    X    |    X    |    X    |
  ---------------------------------------------

< 1 0 1 1 0 1 0 β β
  ^
```

Tendo como entrada “1001110110001”, o resultado seria:

```
     |    0    |    1    |    <    |    β    |
  ---------------------------------------------
  000 | 000,1,D | 000,0,D |    X    | 001,β,E |
  ---------------------------------------------
  001 | 001,0,E | 001,1,E | 002,<,D |    X    |
  ---------------------------------------------
> 002 |    X    |    X    |    X    |    X    |
  ---------------------------------------------

< 0 1 1 0 0 0 1 0 0 1 1 1 0 β β
  ^
```

Decisões de Projeto para a Implementação

O código foi feito utilizando a linguagem de programação Java, são múltiplas as razões para essa escolha, como sua característica de ser multiplataforma, o que facilita muito a execução do código em outros sistemas operacionais. O fato de ser uma linguagem orientada a objetos com sintaxe familiar reduz consideravelmente o tempo de desenvolvimento do programa, em particular para códigos que possuem partes com funcionamento similar, podendo então reutilizar alguns objetos.

Algumas propriedades de herança de classes em Java, em particular o escopo das variáveis, é muito útil para compartilhar informações de controle que várias partes do código precisam ter acesso para manter um funcionamento inteligente.

Para o armazenamento da tabela de função de transição foi implementado um array bidimensional, em que cada célula do array possui variáveis presentes na estrutura padrão de uma função de transição de uma máquina de Turing conceitual, como por exemplo o próximo estado, novo símbolo na fita e a direção do cabeçote.

Foram adicionadas também algumas variáveis de controle para tornar o funcionamento da máquina inteira mais eficiente. Tais variáveis controlam o movimento da máquina como um todo, dando forma ao algoritmo passado para a máquina pela completa descrição formal.

O algoritmo principal do programa, que permite o funcionamento da máquina de Turing, foi desenvolvido considerando as regras de funcionamento de uma máquina padrão, que já foram descritas em alguns parágrafos acima. Assim como uma máquina de Turing conceitual, o algoritmo mantém o estado atual de acordo com o estado inicial e as mudanças feitas respeitando a tabela de função de transição, juntamente com o símbolo da fita. Para cada estado, o algoritmo testa todos os símbolos, armazenados nos alfabetos que o usuário digitou, até encontrar uma transição no estado atual com o mesmo símbolo presente na atual posição do cabeçote na fita.

Uma vez encontrada uma transição, o algoritmo suspende a verificação de símbolos e executa ações usando as variáveis armazenadas na tabela de função de transição, acessando precisamente a célula que corresponde ao estado atual com o símbolo da fita. Tais ações são cruciais para o movimento da máquina, então o algoritmo atualiza o estado atual, escreve um símbolo na fita e movimenta o cabeçote na fita para a esquerda ou direita. Os dados presentes na tabela de função de transição são alimentados pelo usuário antes do algoritmo principal ser executado, só então o algoritmo sendo descrito poderá fazer alterações na fita.

Após tais ações serem executadas, o algoritmo recomeça a procurar por algum símbolo nos alfabetos que seja igual ao símbolo presente na posição atual do cabeçote na fita. Esse loop vai repetir até que o algoritmo encontre algum estado que não possua uma transição com o símbolo da fita, e quando isso acontecer com o estado atual sendo um estado final, que já foi descrito pelo usuário, a máquina suspende seu funcionamento, e aceita a palavra se for uma máquina do tipo reconhecedora, ou simplismente suspende e mostra a fita, se for uma máquina transdutora.

No código atual foi implementada a funcionalidade de mostrar a fita durante o funcionamento da máquina de Turing, portanto todas as alterações feitas na fita, que são as instruções armazenadas na tabela de função de transição, podem ser visualizadas uma de cada vez no momento em que acontecem. Tal funcionalidade é útil para o usuário, em ajudar a diagnosticar erros no algoritmo que por ele está sendo desenvolvido. Juntamente foi implementada também a opção de visualizar a fita apenas no final da execução, pois sem a visualização de cada mudança na fita em tempo real, a máquina de Turing termina sua execução mais rapidamente.

Considerando que a tabela de função de transição pode possuir instruções que, de maneira intencional ou não, permita a máquina de Turing ficar em um loop infinito, foi implementado um contador de transições e também um limite. Portanto quando tal limite de transições é atingido, o programa informa o usuário que várias transições já aconteceram, e pergunta se ele deseja continuar.

Conclusão

Após a experiência de todas as pesquisas, perguntas, testes e resultados que se fizeram presentes no período de desenvolvimento deste trabalho, é certo dizer que o conhecimento sobre a máquina de Turing foi continuamente crescente.

A máquina de Turing certamente é um dispositivo interessante, assim como é o estudo de algoritmos, e o programa que foi desenvolvido e relatado nos parágrafos acima já se provou útil. Com o crescimento estável do poder computacional que os computadores estão mantendo atualmente, os vários campos de pesquisa que fazem proveito dessas máquinas também recebem um aumento em desempenho e precisão nos resultados.

Considerando a grande quantidade de usuários de computadores e serviços básicos que fazem uso do mesmo, novos meios de aplicar as atuais tecnologias estão sendo explorados, também experimentando com diferentes arquiteturas para circunstâncias particulares, soluções algorítmicas mais eficientes continuam sendo procuradas.

REFERÊNCIAS BIBLIOGRÁFICAS

RUSSELL, Stuart; NORVIG, Peter – **Artificial Inteligence** – A Modern Approach – Pearson, 2010.

WIKIPEDIA – **Algorithm**, 2017. [Internet] Disponível em: <https://en.wikipedia.org/wiki/Algorithm>. Acesso em: 16/07/2017

WIKIPEDIA – **Computability**, 2017. [Internet] Disponível em: <https://en.wikipedia.org/wiki/Computability>. Acesso em: 16/07/2017

WIKIPEDIA – **Computational problem**, 2017. [Internet] Disponível em: <https://en.wikipedia.org/wiki/Computational_problem>. Acesso em: 16/07/2017

WIKIPEDIA – **Automata theory**, 2017. [Internet] Disponível em: <https://en.wikipedia.org/wiki/Automata_theory>. Acesso em: 16/07/2017

WIKIPEDIA – **Formal language**, 2017. [Internet] Disponível em: <https://en.wikipedia.org/wiki/Formal_language>. Acesso em: 16/07/2017

WIKIPEDIA – **Discrete mathematics**, 2017. [Internet] Disponível em: <https://en.wikipedia.org/wiki/Discrete_mathematics>. Acesso em: 16/07/2017

WIKIPEDIA – **Alan Turing**, 2017. [Internet] Disponível em: <https://en.wikipedia.org/wiki/Alan_Turing>. Acesso em: 16/07/2017

WIKIPEDIA – **Embedded system**, 2017. [Internet] Disponível em: <https://en.wikipedia.org/wiki/Embedded_system>. Acesso em: 16/07/2017